

---

# Desarrollo de una agenda escolar predictiva para niños con trastorno del espectro autista

---



Trabajo de Fin de Grado  
Curso 2018–2019

Autor

Mónica Hernández Gómez  
Iván García Baños

Director

Gonzalo Méndez Pozo  
Susana Bautista Blanco

Grado en Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid



# Desarrollo de una agenda escolar predictiva para niños con trastorno del espectro autista

**Trabajo de Fin de Grado en Ingeniería Informática**  
**Departamento de Ingeniería del software e inteligencia**  
**artificial**

## **Autor**

**Mónica Hernández Gómez**  
**Iván García Baños**

## **Director**

**Gonzalo Méndez Pozo**  
**Susana Bautista Blanco**

*Dirigida por el Doctor*

**Gonzalo Méndez Pozo**  
**Susana Bautista Blanco**

**Grado en Ingeniería Informática**  
**Facultad de Informática**  
**Universidad Complutense de Madrid**

**20 de septiembre de 2019**



# Agradecimientos

En primer lugar, queremos agradecer a nuestros directores habernos dado la oportunidad de poder realizar este proyecto. También queremos dar las gracias a todos los profesores que nos han acompañado y enseñado durante la carrera motivándonos para llegar a este momento.

También queremos agradecer a nuestras familias y amigos por habernos apoyado y animado en los momentos difíciles y haber estado siempre a nuestro lado. Agradecemos a nuestros familiares todos aquellos consejos que nos han ido dando, junto a su comprensión y fuerza con las que nos han impregnado. También les damos gracias por la confianza que han puesto en nosotros en todo momento, sobre todo cuando nosotros mismos la perdíamos.

Les damos gracias a nuestros compañeros y amigos, que también nos han dado su apoyo y que nos han ayudado a divulgar esta aplicación para que llegue al mayor número de personas posible.

Por último también queríamos darles las gracias a todas aquellas personas que han colaborado con nosotros, realizando las pruebas y divulgando también la aplicación a sus conocidos.

¡Muchas gracias a todos!



# Resumen

En la actualidad gracias al avance de internet y las tecnologías cada vez hay mayor cantidad de información disponible y mayor número de personas tienen acceso a un sistema de comunicación sencilla. Pero, a pesar de estos avances, aún hay determinadas personas que no se pueden beneficiar de ellos.

En nuestro caso vamos a desarrollar una aplicación web dirigida a personas con Trastorno del Espectro Autista (TEA). Estas personas tienen dificultades de comunicación oral y escrita, y para solventar dichas dificultades vamos a utilizar Sistemas Aumentativos y Alternativos de la Comunicación (SAACs). En nuestro caso, hemos elegido hacer uso de pictogramas para representar parte de la información que habrá en la aplicación.

Para diseñar la aplicación hemos utilizado el diseño centrado en el usuario. Éste está guiado por la información acerca de las personas que van a utilizar la aplicación. En este tipo de diseño prevalece el usuario final sobre otros factores en la toma de decisiones. También hemos realizado un análisis de las aplicaciones similares ya existentes para tener una perspectiva inicial, y ver que funcionalidades hay implementadas y cuales no, para a partir de esa base, desarrollar nuestro proyecto.

Posteriormente hemos realizado el diseño de la interfaz basándonos en las pantallas que hemos visto en otras aplicaciones con un propósito similar. Los diseños se han realizado teniendo en cuenta cuales son las necesidades de las personas con Trastorno del Espectro Autista (TEA) y siguiendo los principios del diseño centrado en el usuario.

Por último hemos realizado evaluaciones heurísticas para conocer los aspectos a mejorar con el feedback recibido por los usuarios que han probado la aplicación. También se analizarán las conclusiones a las que se ha llegado una vez finalizado el proyecto y quedarán ideas respecto al trabajo futuro que se pueda realizar para continuar mejorando este trabajo de fin de grado.

## **Palabras clave**

Trastorno del Espectro Autista (TEA), Predictor, Modelos de Markov, Pictogramas, Sistemas Aumentativos y Alternativos de la Comunicación (SAAC).



# Abstract

Nowadays, thanks to the advancement of the Internet and technologies, more and more information is available and more and more people have access to a simple communication system. But despite these advances, there are still some people who cannot benefit from these advances.

In our case we are going to develop a web application aimed at people with Autism Spectrum Disorder (ASD). These people have oral and written communication difficulties, and to solve these difficulties we are going to use Augmentative and Alternative Communication Systems (AAC). In our case we will use pictograms to show the information.

We used user-centered design to design the application. This is guided by information about the people who will use the application. In this type of design the end user prevails over other factors in decision making. We have also carried out an analysis of similar applications in order to have a perspective on the existing applications and the functionalities they provide.

We then designed the interface based on others with a similar purpose, taking into account the needs of people with ASD and following the principles of user-centered design.

And lastly, we have carried out heuristic evaluations in order to know the aspects to be improved. We have also analyzed the conclusions of the project and extracted the future work that can be done from this tfg.

## Keywords

autistic spectrum disorders (ASD), Predictor, Markov Models, Pictograms, Augmentative and Alternative Communication (AAC).



# Índice

<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Goals . . . . .	2
1.3. Project structure . . . . .	2
<b>1. Introducción</b>	<b>5</b>
1.1. Motivación . . . . .	5
1.2. Objetivos . . . . .	6
1.3. Estructura del proyecto . . . . .	6
<b>2. Estado de la Cuestión</b>	<b>9</b>
2.1. Trastorno del Espectro Autista . . . . .	10
2.1.1. Autismo en las aulas . . . . .	11
2.1.2. Beneficios del uso de TICs en personas con TEA . . . . .	12
2.2. Sistemas Aumentativos y Alternativos de la Comunicación (SAACs) . . . . .	13
2.3. Pictogramas . . . . .	14
2.3.1. Pictogramas de ARASAAC . . . . .	14
2.3.2. Aplicaciones basadas en pictogramas . . . . .	15
2.4. Predictores . . . . .	21
2.4.1. Modelos de Markov . . . . .	21
2.4.2. Cadena de Markov . . . . .	23
2.4.3. Ngramas . . . . .	25
2.5. Diseño Centrado en el Usuario . . . . .	25
2.5.1. Usabilidad . . . . .	26
2.6. Conclusiones . . . . .	32
<b>3. Diseño de la interfaz</b>	<b>33</b>
3.1. Diseño competitivo . . . . .	33

3.2.	Primera versión . . . . .	34
3.2.1.	Versión de Mónica . . . . .	34
3.2.2.	Versión de Iván . . . . .	35
3.3.	Segunda versión . . . . .	35
3.3.1.	Escoger ideas de ambas versiones . . . . .	36
3.3.2.	Página de inicio . . . . .	37
3.3.3.	Menú desplegable . . . . .	38
3.3.4.	Inicio . . . . .	38
3.3.5.	Composiciones . . . . .	38
3.3.6.	Calendario . . . . .	39
3.3.7.	Cumpleaños . . . . .	39
3.4.	Primera versión implementada . . . . .	41
3.5.	Versión final . . . . .	41
3.5.1.	Página principal . . . . .	42
3.5.2.	Mi Perfil . . . . .	44
3.5.3.	Panel de administración . . . . .	45
3.6.	Conclusiones . . . . .	45
<b>4.</b>	<b>Tecnologías</b>	<b>47</b>
4.1.	HTML . . . . .	47
4.2.	CSS . . . . .	48
4.3.	XAMPP . . . . .	49
4.3.1.	Apache . . . . .	49
4.3.2.	MySQL . . . . .	49
4.3.3.	PHP . . . . .	50
4.4.	GitHub . . . . .	50
4.5.	JavaScript . . . . .	50
4.6.	jQuery . . . . .	51
4.7.	Ajax . . . . .	52
4.8.	REST . . . . .	52
4.9.	SQL . . . . .	53
4.10.	Bootstrap . . . . .	53
4.11.	JSON . . . . .	53
4.12.	Servicios Web . . . . .	54
4.13.	Java . . . . .	54
4.14.	Jersey . . . . .	55
4.15.	GNU/Linux . . . . .	56
4.16.	Conclusiones . . . . .	56
<b>5.</b>	<b>PictoPad</b>	<b>57</b>
5.1.	Modelo-Vista-Controlador (MVC) . . . . .	57

5.2.	Bases de Datos . . . . .	58
5.3.	PictoPad . . . . .	59
5.4.	Sección Traductor . . . . .	60
5.5.	Sección Editor . . . . .	61
5.6.	Sección Buscador . . . . .	62
5.7.	Sección Sugerencias . . . . .	63
5.8.	Perfil de usuario . . . . .	63
5.9.	Panel de administración . . . . .	64
5.10.	Predictor . . . . .	64
5.10.1.	Predictor basado en análisis de frecuencias . . . . .	65
5.10.2.	Predictor basado en Cadenas de Markov . . . . .	65
5.10.3.	Implementación del predictor . . . . .	65
5.10.4.	Creación de servicios REST . . . . .	68
5.11.	Conclusiones . . . . .	68
<b>6.</b>	<b>Pruebas</b>	<b>69</b>
6.1.	Evaluación de la aplicación . . . . .	69
6.1.1.	Evaluación heurística . . . . .	70
6.1.2.	Aspectos positivos . . . . .	72
6.2.	Evaluación de los predictores . . . . .	72
6.2.1.	Resultados . . . . .	72
6.3.	Conclusiones . . . . .	73
<b>7.</b>	<b>Trabajo individual</b>	<b>75</b>
7.1.	Trabajo realizado por Mónica . . . . .	75
7.2.	Trabajo realizado por Iván . . . . .	77
<b>8.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>79</b>
8.1.	Conclusiones . . . . .	79
8.2.	Trabajo Futuro . . . . .	81
<b>8.</b>	<b>Conclusions and Future Work</b>	<b>83</b>
8.1.	Concluding Remarks . . . . .	83
8.2.	Future Work . . . . .	84
	<b>Bibliografía</b>	<b>87</b>



# Índice de figuras

2.1. Tríada de Wing . . . . .	10
2.2. Pictograma que representa la palabra Solidaridad . . . . .	15
2.3. De izquierda a derecha se observan los pictogramas cocinar a color y en blanco y negro y jugar al fútbol de tipo esquemático	15
2.4. Imagen que muestra las herramientas online de ARASAAC . . . . .	16
2.5. Pictoagenda. <a href="https://www.pictoagenda.com/">https://www.pictoagenda.com/</a> . . . . .	17
2.6. Pictar. <a href="http://hypatia.fdi.ucm.es/pictar/">http://hypatia.fdi.ucm.es/pictar/</a> . . . . .	18
2.7. Pictotraductor. <a href="https://www.pictotraductor.com/">https://www.pictotraductor.com/</a> . . . . .	19
2.8. Editor predictivo de mensajes . . . . .	20
2.9. Probabilidades de transición de los estados . . . . .	24
3.1. Vista diseño inicial de Mónica . . . . .	34
3.2. Vista diseño inicial de Iván . . . . .	36
3.3. Vista página de inicio versión 3 . . . . .	37
3.4. Vista menú versión 3 . . . . .	38
3.5. Vista página Composiciones versión 3 . . . . .	39
3.6. Vista calendario versión 3 . . . . .	40
3.7. Vista página cumpleaños versión 3 . . . . .	40
3.8. Vista de la primera versión implementada . . . . .	41
3.9. Vista de la página principal de la versión final . . . . .	43
3.10. Vista de la página Mi Perfil de la versión final . . . . .	44
3.11. Vista de la página Administración de la versión final . . . . .	45
5.1. Modelo-Vista-Controlador . . . . .	58
5.2. Diagrama representativo de la Base de Datos . . . . .	59
5.3. Diagrama de actividad de la sección Traductor . . . . .	61
5.4. Diagrama de actividad de la sección Buscador . . . . .	63
5.5. Diagrama representativo del predictor diseñado por Mónica . . . . .	66





# Índice de tablas



# Chapter 1

## Introduction

In this chapter we are going to make an introduction about the work that has been done. In order to do so, we will talk about the motivation that has led us to carry out this project and the objectives that we want to achieve with this work. We will also explain to whom it is geared and the way we have chosen to structure the report.

### 1.1. Motivation

Nowadays, the great advance of Internet and technologies, makes more and more information available and allows a higher number of people to have access to it through a simple and fast communication system. All this is having a strong impact on society.

Despite all these advances, there are people who still cannot overcome the difficulties that these new communication systems have, making it very difficult for them to access information and use the communication tools that are available.

One of the problems encountered by users is that the design of the interfaces used by the Internet is not focused on all types of users. Another problem is that standards are not followed to display information on all websites so that all users are able to access it in a simplified way.

To overcome these difficulties, specialized applications have been developed to help eliminate these difficulties for a certain type of users.

In our case we are going to develop a web application aimed at people with Autism Spectrum Disorder. These people have difficulties of oral and written communication, besides that an excess of information or the great changes in the form to show it can suppose them great difficulties.

To facilitate access to information we are going to use Augmentative and Alternative Communication Systems (AAC), in our case we are going to use

pictograms to show the information.

## 1.2. Goals

The aim of our project is to develop a web application accessible to people with ASD that is useful to facilitate communication. The idea is to provide a translator of text to pictograms and one vice versa, so that communication with other people is easier. Another component will be a pictogram predictor to facilitate the design of phrases from drawings.

In order to achieve the first objective we have set ourselves, we will use the web services developed at the Complutense University for this purpose. The pictograms that we have chosen are those of ARASAAC since it has a wide database and its use is widespread, as we will see later.

For the completion of the predictor we will use Markov Models and Ngrams, which will give us the possibility to find out the next pictogram that a specific user will use and the possibility to customize the prediction for that user.

To design the interface we will focus on the user, so you must get as much information as possible from users, which in our case are people with Autism Spectrum Disorder, in order to adapt to their needs.

We also have to take into account the academic purpose of this work, in which the main objective is to demonstrate the knowledge learned during the degree in addition to deepen in certain concepts that are necessary for the completion of this work.

If we achieve the objectives described above we can provide a simpler form of communication to people who make use of pictograms, which can be used both in the family, as well as social and academic.

## 1.3. Project structure

The report of our work consists of eight chapters. In addition there is another one, the bibliography, in which the sources from where the information has been extracted are referenced.

The first chapter, Introduction, explains the motivation for carrying out the project, the objectives to be achieved and a brief summary of the contents that will be dealt with in the following chapters. The introduction will be in both Spanish and English.

In the second chapter, State of the question, we will talk about Autism Spectrum Disorder (ASD), indicating what it is and the benefits for people with ASD from the use of technology. We will also talk about pictograms and the main applications that make use of them. At the end, we will talk about user-centered design, indicating what it is and what it involves.

---

Third chapter, Interface Design, shows the process of designing the GUI using user-centered design methods.

Fourth chapter talks about Technologies. There, the technologies used for the development of this project are shown.

The fifth chapter will deal with internal architecture. It will explain how the design has been and will deal more in depth with the most important parts of the implementation of the web. It will also consider everything related to the predictor, whether the designs as well as the algorithms that are created.

The sixth chapter covers the tests carried out to check the correct functioning of the application. The results of the heuristic tests will also be presented.

In chapter number seven, each member of the team will present what kind of things has done on the project.

Finally, in the eighth chapter, the conclusions obtained after the realization of the project will be exposed. It will be also detailed what are the ideas about how this project could be continued in the future. This section will be written in both Spanish and English.



# Capítulo 1

## Introducción

*“Every adventure requires a first step”*

— The Chesire cat

En este capítulo vamos a realizar una introducción sobre el trabajo desarrollado, detallando la motivación que nos ha llevado a efectuar este proyecto, los objetivos que queremos alcanzar con dicho trabajo, a quién va dirigido y la forma que hemos elegido para estructurar la memoria.

### 1.1. Motivación

Actualmente, el gran avance de Internet y las tecnologías hace que cada vez haya mayor cantidad de información disponible y permite que un mayor número de personas tenga acceso a ella mediante un sistema de comunicación sencillo y rápido.

Pese a todos estos avances, hay personas que aún no pueden salvar las dificultades que estos nuevos sistemas de comunicación tienen, haciéndoles muy complicado el acceso a la información y el uso de las herramientas de comunicación que se encuentran disponibles.

Uno de los problemas con los que se encuentran los usuarios es que el diseño de las interfaces que se utilizan en Internet no están enfocadas a todos los tipos de usuarios, especialmente a los que sufren algún tipo de discapacidad o de un rango de edad determinado. Otro problema es que no se sigan unos estándares para mostrar la información en todos los sitios web de forma que todos los usuarios sean capaces de acceder a ella de forma simple.

Para salvar estas dificultades se han desarrollado aplicaciones especializadas que ayudan a eliminar dichas dificultades para un determinado tipo de usuarios.

En nuestro caso vamos a desarrollar una aplicación web dirigida a personas con Trastorno del Espectro Autista. Estas personas tienen dificultades de

comunicación oral y escrita, además, un exceso de información o los grandes cambios en la forma de mostrarla les puede suponer enormes dificultades.

Para facilitar el acceso a la información vamos a utilizar Sistemas Aumentativos y Alternativos de la Comunicación (SAACs), en nuestro caso vamos a utilizar pictogramas para mostrar la información.

## 1.2. Objetivos

El objetivo de nuestro proyecto es desarrollar una aplicación web accesible para personas con TEA y que ésta les sea de utilidad para facilitarles la comunicación. Para ello la idea es proporcionarles un traductor de texto a pictogramas y pictogramas a texto indistintamente. Otro de los componentes será un predictor de pictogramas para facilitar el diseño de frases a partir de dibujos.

Para alcanzar el primer objetivo que nos hemos fijado, utilizaremos los servicios web desarrollados en la Universidad Complutense para este fin, en concreto los dedicados a traducción de texto y obtención de pictogramas. Los pictogramas que hemos escogido son los de ARASAAC ya que se dispone de una amplia base de datos y su uso está muy extendido, como veremos mas adelante.

Para la realización del predictor utilizaremos Modelos de Markov y Ngramas, lo que nos dará la posibilidad de averiguar el siguiente pictograma que usará un usuario concreto y la posibilidad de personalizar la predicción para ese usuario.

Para diseñar la interfaz nos vamos a centrar en el usuario, por lo que se debe obtener la mayor cantidad de información posible de los usuarios, que en nuestro caso son personas con Trastorno del Espectro Autista, para poder adaptarnos a sus necesidades.

También tenemos que tener en cuenta la finalidad académica de este trabajo, en la que el principal objetivo es demostrar los conocimientos aprendidos durante el grado además de profundizar en determinados conceptos que son necesarios para la realización de este trabajo.

Si alcanzamos los objetivos que hemos descrito anteriormente podremos proporcionar una forma mas sencilla de comunicación a las personas que hagan uso de pictogramas, pudiendo usarse tanto en el ámbito familiar, como social y académico.

## 1.3. Estructura del proyecto

La memoria de nuestro trabajo consta de ocho capítulos además de otro de bibliografía en el que se referencian las fuentes de las que se ha extraído la información.



En el primer capítulo, Introducción se explican la motivación para realizar el proyecto, los objetivos que se pretenden alcanzar y un breve resumen de los contenidos que se abordarán en los capítulos siguientes. La introducción se realizará tanto en castellano como en inglés.

En el segundo capítulo, Estado de la cuestión hablaremos del Trastorno del Espectro Autista (TEA) indicando qué es y los beneficios para las personas con TEA del uso de la tecnología. También hablaremos sobre los pictogramas y las principales aplicaciones que hacen uso de ellos. Por último hablaremos acerca del diseño centrado en el usuario, indicando qué es y en qué consiste.

En el tercer capítulo llamado Diseño de la interfaz se muestra el proceso de diseño de la interfaz gráfica siguiendo los métodos del diseño centrado en el usuario.

En el cuarto capítulo, Tecnologías se muestran las tecnologías utilizadas para la elaboración de este proyecto.

El quinto capítulo tratará sobre la arquitectura interna. En él se explicará como ha sido el diseño y se tratarán más a fondo las partes más importantes de la implementación de la web. También contempla todo lo relacionado con el predictor, tanto los diseños como los algoritmos que se creen.

El sexto capítulo engloba las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación. En él también se expondrán los resultados de las pruebas heurísticas.

En el capítulo séptimo, cada miembro del equipo expondrá el trabajo que ha realizado en el proyecto.

Por último, en el capítulo octavo, se expondrán las conclusiones obtenidas tras la realización del proyecto. También se detallará cuales son las ideas sobre como podría continuarse con este proyecto en el futuro. Este capítulo se realizará tanto en castellano como en inglés.



## Capítulo 2

# Estado de la Cuestión

*“Tell me, and I forget. Teach me, and I remember. Involve me, and I learn.”*

— Benjamin Franklin

Para comenzar a desarrollar este proyecto, el primer objetivo que nos planteamos es tratar de comprender cuales son los problemas y las necesidades específicas que tienen las personas con trastorno del espectro autista (en adelante TEA). Para ello comenzamos con la búsqueda de recursos digitales que se hallen actualmente desarrollados y en funcionamiento. Partiendo como base de estos conocimientos, pretendemos desarrollar nuestro trabajo para solucionar algunas de las carencias que actualmente existen en estos sistemas.

Primeramente realizaremos una pequeña introducción sobre el tema en el que se definan los TEA, los Sistemas Aumentativos y Alternativos de la Comunicación (SAACs), los pictogramas y las aplicaciones que hacen uso de este SAAC.

Como se ha mencionado en los objetivos de este trabajo (ver apartado 1.2), pretendemos dotar a nuestra aplicación de un mecanismo de predicción. Por este motivo también estudiaremos los mecanismos que hoy en día se utilizan de formas más amplia en las predicciones de texto.

Seguidamente hablaremos del enfoque que hemos dado a la interacción con los usuarios. Esto lo explicaremos en el apartado *Diseño Centrado en el Usuario*. En esta parte también se explicarán las necesidades del usuario en las que nos hemos centrado.

Ya que todo ello representa la base de nuestro proyecto, es importante poner en situación lo que representa cada tema, para que al mencionarse en capítulos posteriores cualquiera de estos términos, se pueda identificar con claridad a que corresponde.

## 2.1. Trastorno del Espectro Autista

El trastorno del espectro autista, también conocido como TEA, se define como una disfunción neurológica crónica que puede aparecer desde edades tempranas con una serie de síntomas que se manifiestan en la interacción social, la comunicación y en la capacidad de razonamiento durante la interacción social, acompañado de la necesidad de una rutina diaria en la que se experimenten los mínimos cambios posibles. Estos síntomas se conocen como triada de Wing.

La triada de Wing representa las tres áreas que se encuentran perjudicadas por el Síndrome de Asperger (tipo de trastorno incluido en los TEA). (Jaume Guilera, 2016)

- **Área social:** representa la dificultad que tienen las personas con éste síndrome de poder expresarse, ya que no tienen las facultades de poder interpretar los sentimientos tanto suyos como de terceros. Esto hace que sea dificultosa la comunicación y que se puedan dar reacciones disformes en cuanto a los sentimientos.
- **Área comunicativa:** aunque no muestren problemas de vocabulario, si que carecen de un lenguaje expresivo tanto en la comunicación verbal como en la no verbal. Esto hace que su comunicación sea mayormente uniforme.
- **Área psicomotora:** son personas que presentan una débil armonización, lo cuál hace que sus actividades y movimientos sean iterativas.

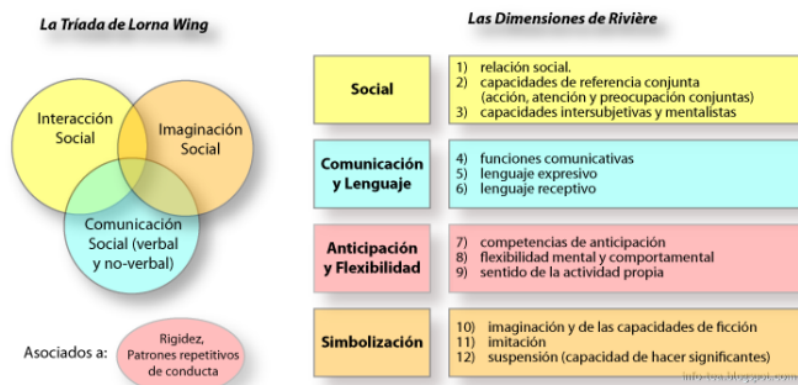


Figura 2.1: Triada de Wing

La gravedad, la forma y la edad de los criterios varían dependiendo de cada individuo y definen las categorías del TEA. (Grupo de Trabajo de la

Guía de Práctica Clínica para el Manejo de Pacientes con Trastornos del Espectro Autista en Atención Primaria, 2009).

Existen varios tipos de trastornos, que se explican brevemente a continuación:

**Síndrome de Kanner (Autismo)** Las personas con dicho síndrome presentan una conexión emocional con su entorno más limitada, haciendo que parezca que se encuentran en su propio mundo. Se caracterizan por tener comportamientos reiterados y por su alta agitación respecto a estímulos externos como son las luces o algunos sonidos.

**Síndrome de Asperger** Este tipo de trastorno se caracteriza por su difícil diagnóstico, ya que se suele presentar en personas con una inteligencia media o alta. Su carencia está en sus competencias sociales, lo que les impide una completa integración con la sociedad y en los entornos laborales debido a su falta de empatía, escasa coordinación o no entender la ironía o el uso del doble sentido del lenguaje.

**Síndrome de Heller** El síndrome de Heller, también conocido como trastorno desintegrador infantil, afecta a las mismas partes que el resto de síndromes: lenguaje, motricidad e inclusión social. La diferencia respecto a los anteriores es que es regresivo y repentino. Las personas que padecen este tipo de síndrome son capaces de percatarse del problema que tienen.

**Trastorno generalizado del desarrollo no especificado** Las personas que padecen el trastorno autista pero no encajan totalmente con ninguno de los tipos anteriores. Estas personas se caracterizan por su déficit de intercambio o comunicación social, además de por sus peculiaridades en sus intereses o actividades.

### 2.1.1. Autismo en las aulas

El autismo es un trastorno que hasta el momento carece de una cura, pero lo que sí existe es un tratamiento que mejora su desarrollo y el bienestar de los niños. En las aulas, el uso de terapias estimulantes junto a una buena educación y a la integración social supervisada es muy importante para el desarrollo de estos niños (Universidad Internacional de Valencia, 2014).

Un elemento muy útil son las pizarras digitales para ayudar en el aprendizaje, ya que los contenidos se presentan con un mayor tamaño y los hace más atractivos. Se pueden utilizar aplicaciones especializadas y contenidos adaptados, como los pictogramas para intentar conseguir que el aprendizaje se realice de forma más amena para los alumnos.

Existen aplicaciones y programas específicos para ayudar a los niños con TEA a profundizar en los contenidos necesarios para su aprendizaje y potenciar específicamente sus habilidades educativas y sociales, además de mejorar su autonomía personal.

Un recurso importante es el material informático adaptado, como los teclados, que tienen las teclas diferenciadas por colores dependiendo de la funcionalidad.

### **2.1.2. Beneficios del uso de TICs en personas con TEA**

Igual que para la educación en general, la utilización de las TIC (tecnologías de la información y comunicaciones) es muy útil para ayudar en el aprendizaje de alumnos con necesidades educativas especiales, como puedan ser en este caso los alumnos con TEA. Su uso permite que niños que presentan este tipo de trastorno puedan desarrollar de una forma más eficaz el lenguaje.

Según la Confederación Autismo de España, el uso de las TICs hace más sencillo que las personas con TEA se integren, y mejora sus habilidades en el aprendizaje haciendo que tengan una mayor comprensión del lenguaje. Además éstas ayudan a que estas personas puedan desarrollar capacidades para desenvolverse de forma menos dependiente y así acrecentar la realización de tareas cotidianas de forma autónoma. (Confederación Autismo España, 2018)

En los últimos años, gracias a la introducción de Internet y las nuevas tecnologías en la vida cotidiana, han aparecido numerosas formas de ayudar a los niños con TEA. Estas mejoras han tenido lugar tanto en las aulas, con la incorporación de pizarras digitales, como en la vida diaria, con aplicaciones basadas en pictogramas o videojuegos educativos.

El uso de las pizarras interactivas en las clases es beneficioso a la hora de presentar todos los contenidos, ya que los muestra de una forma más sencilla, ingeniosa y atractiva. Por otro lado, las aplicaciones permiten que los niños con TEA aceleren su aprendizaje y además adquieran y mejoren las habilidades que tienen. Con el uso de estas aplicaciones, no solo ganan en el ámbito educativo, si no que les permiten desarrollar mejoras en su autonomía para la vida cotidiana. Para poder hacer uso de todo ello, también se han hecho mejoras en los materiales necesarios para poder manejar estas aplicaciones como son los teclados, en los que se reducen el número de teclas que tienen lo que permite que estas puedan ser más grandes y visuales. (Universidad Internacional de Valencia, 2014)

La mayoría de estas aplicaciones hacen uso de los pictogramas, ya que para los niños es más sencillo asociar los distintos conceptos que van aprendiendo a través del uso de imágenes, ya que como dice el refrán, “una imagen vale más que mil palabras”.

## 2.2. Sistemas Aumentativos y Alternativos de la Comunicación (SAACs)

Los SAACs son formas de comunicación distintas al lenguaje hablado cuyo objetivo es compensar las dificultades que presenta el lenguaje hablado para las personas con discapacidad (Basil, 2018) .

Un ejemplo son las personas que tienen una discapacidad visual y utilizan el Braille. Este, es un sistema aumentativo que permite la lectura y escritura a personas ciegas, y que consiste en una combinación de puntos en relieve agrupados en cédulas que representan letras, números o símbolos matemáticos.

Los SAACs se clasifican en dos categorías, dependiendo de si necesitan ayuda externa o no. Dentro de cada una de estas categorías hay gran cantidad de sistemas.

Los sistemas sin ayuda solo utilizan el propio cuerpo y no necesitan usar ningún material adicional. En esta categoría destacan el habla y el lenguaje de signos que utilizan las personas con problemas auditivos (sólo utiliza las manos y expresiones faciales).

Los sistemas con ayuda hacen uso de recursos externos al propio cuerpo como sistemas de símbolos abstractos (pictogramas), puntos en relieve (Braille) o recursos informáticos (apps, páginas web, etc).

Estos sistemas los utilizan principalmente personas con dificultades en la comunicación, ya que están diseñados para facilitar el aprendizaje y la comprensión en personas con TEA o enfermedades neurológicas como la parálisis cerebral, esclerosis lateral amiotrófica o párkinson. No solo lo utilizan personas que presentan dificultades de comunicación, si no que también están pensados para que los usen personas que se comunican con lenguaje de signos, ya que hacen uso de ellos con elementos paralingüísticos como los gestos, las expresiones faciales o símbolos que representan conceptos.

Como su nombre indica, los SAACs hacen uso de la comunicación aumentativa y alternativa (CAA), que abarca todo tipo de sistemas simbólicos, incluyendo fotografías, palabras, pictogramas, mímico y gestos entre otros tipos de símbolos que se dividen en tres principales grupos:

- **Gestuales** Se corresponde con la gesticulación de las personas y en ello se incluye tanto los gestos comunes como los manuales que puede llegar a crear una persona cuando se comunica con los demás. Este tipo lo utilizan las personas con TEA y con alguna discapacidad intelectual, ya que poseen una capacidad motriz suficiente para ello.
- **Gráficos** Este tipo está formado por imágenes y vocablos. Lo emplean aquellas personas que presentan discapacidades motrices, además de las citadas en el apartado anterior.

- **Pictográficos** Lo utilizan aquellas personas que no están instruidas, ya sea por alguna enfermedad que las discapacite, o por la edad que tienen. A pesar de que no se consiga un lenguaje totalmente completo, sí que se puede llegar a alcanzar un nivel de comunicación audaz y evolucionado.

## 2.3. Pictogramas

Un pictograma es una imagen o símbolo que representa un objeto del mundo real, una idea, un sentimiento, una acción, etc. Estas imágenes deben ser claras, sencillas y fáciles de comprender (ver Figura 2.2).

Hoy en día, los pictogramas se encuentran muy extendidos. Tanto es así, que están presentes en la mayoría de lugares públicos como son los hospitales, colegios, etc. Estas imágenes se usan de forma que sustituyan al texto escrito ya que expresan los conceptos de una forma más visual, lo que hace que el número de personas que lo comprenda sea superior. Pueden ser utilizados como sistemas alternativos de comunicación, lo cual motiva que sus principales usuarios suelen ser personas con dificultades en la comunicación oral y/o escrita (Wikipedia, 2018b).

En la actualidad no hay definido un estándar para desarrollar pictogramas, pero los más extendidos son los que se pueden encontrar en ARASAAC, de los cuales hablaremos en la siguiente sección.

Hay distintos tipos de pictogramas que se diferencian en cómo representan los conceptos. Los principales modelos son los siguientes: (Delgado, 2012)

- **Color:** en los que se representa el concepto con colores(ver Figura 2.3).
- **Sin color:** en los que se muestra solamente la silueta del pictograma en blanco y negro (ver Figura 2.3).
- **Esquemáticos**, en los que se suelen representar acciones y están basados en siluetas (ver Figura 2.3).
- **Contraste en blanco y negro**, están dirigidos a personas con problemas de visión.

### 2.3.1. Pictogramas de ARASAAC

El Portal Aragonés de Comunicación Aumentativa y Alternativa (ARASAAC)<sup>1</sup>, se creó en el año 2007 entre un grupo de trabajo del Centro Aragonés de Tecnologías para la Educación (CATEDU) y el Colegio

---

<sup>1</sup><http://www.arasaac.org/>





Figura 2.2: Pictograma que representa la palabra Solidaridad



Figura 2.3: De izquierda a derecha se observan los pictogramas cocinar a color y en blanco y negro y jugar al fútbol de tipo esquemático

Público de Educación Especial Alborada, con el objetivo de crear un sistema pictográfico de libre distribución con la intención de facilitar la comunicación a las personas que lo puedan necesitar.

En la actualidad estos pictogramas son los mas utilizados por la mayoría de aplicaciones que hacen uso de ellos.

La base de datos cuenta con mas de 18000 pictogramas en color, 16000 en blanco y negro, casi 1500 fotografías, además de gran cantidad de vídeos en lengua de signos española.

### 2.3.2. Aplicaciones basadas en pictogramas

A partir de los pictogramas, de los que hemos hablado en la sección anterior, han surgido aplicaciones basadas en ellos destinadas a facilitar la comunicación a las personas que padecen algún tipo de discapacidad o que no están alfabetizadas debido a la edad.

A continuación vamos a hablar sobre algunas de estas aplicaciones.

**ARASAAC** Cómo ya hemos expuesto anteriormente, es el Portal Aragonés de Comunicación Aumentativa y Alternativa. En él se pueden encontrar gran cantidad de herramientas en línea para crear contenido adaptado. También tiene a disposición de los usuarios software que simplifica el acceso a las TICs a aquellos que presentan algún tipo de discapacidad.

Las herramientas en línea permiten crear materiales explotando todos los recursos que se encuentran disponibles en la web de ARASAAC. Las principales funcionalidades que nos ofrece esta página, son modificar o crear pictogramas, y traducir frases escritas en lenguaje natural a estos. De la misma forma, nos permite crear horarios, calendarios y tableros. Además, tiene herramientas con las que se pueden crear juegos como la oca, dominó o bingo.

El principal beneficio del software que nos ofrece esta página web, es que favorece el acceso a Internet y a los ordenadores. Nos permite el uso de agendas visuales, aplicaciones de comunicación aumentativa, educativas y de mensajería instantánea, además de editores gráficos y procesadores de texto con pictogramas (ver Figura 2.4).

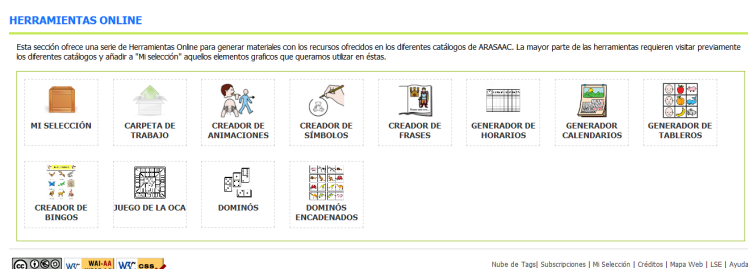


Figura 2.4: Imagen que muestra las herramientas online de ARASAAC

**Pictoagenda** Es una aplicación web diseñada para crear actividades diarias utilizando pictogramas. Con la agenda se ayuda a organizar mentalmente las actividades de cada día evitando cambios inesperados que puedan provocar ansiedad por no comprender las modificaciones que hayan sufrido las actividades(ver Figura 2.5).

Esta aplicación tiene un funcionamiento sencillo que se basa en introducir una palabra o frase para obtener el pictograma relacionado con la actividad buscada. Se basa en un cuadro que tiene escrita la frase por defecto “tu texto”, y que tras escribir la búsqueda, aparece el pictograma relacionado.

La aplicación ofrece la posibilidad de añadir y eliminar pictogramas con los botones “+” y “-” respectivamente. En la parte superior e inferior de la pantalla aparecen dos flechas con las que se puede elegir entre diferentes pictogramas asociados a la actividad referida. En la parte superior del pictograma aparece un cuadro con el que se puede indicar la actividad actual, si se ha podido o no realizar, o si está prohibido. También permite hacer uso de imágenes subidas por el usuario, guardar frases como favoritas, imprimir las composiciones de pictogramas y compartir las frases que desee el usuario a través de redes sociales o co-

reco electrónico. Esta aplicación utiliza los pictogramas de ARASAAC. (pic, 2018a)



Figura 2.5: Pictoagenda. <https://www.pictoagenda.com/>

**PICTAR** Es una aplicación creada por Alejandro Martín en 2018. Con ella se busca facilitar la comunicación para personas que poseen carencias en el lenguaje oral, y que no tienen la capacidad parcial o total de expresarse de forma hablada. (Martín Guerrero, 2018)(ver Figura 2.6).

La principal utilidad que ofrece esta aplicación es la traducción de texto escrito a pictogramas, para así hacer que sea más sencilla la comunicación con el usuario. Se introduce el texto deseado y la aplicación genera los pictogramas asociados a esa frase.

En la parte inferior se puede encontrar un editor que permite copiar los pictogramas generados en la traducción o importar imágenes desde los archivos locales. Una vez que tenemos los pictogramas elegidos, presenta la opción de poderles cambiar el color a blanco y negro para que sean menos llamativos. Las frases que se generen pueden ser exportadas en formato json, algo poco útil a la hora de usar, puesto que una vez descargado, no se pueden ver las imágenes. Con el número de elementos y columnas consigue jugar con el tamaño de los pictogramas y así a menor número de columnas, las imágenes podrán ser mejor visualizadas. También dispone a la derecha de un buscador de pictogramas, para así poderlos usar en el editor sin necesidad de importar imágenes o traducir frases o palabras.

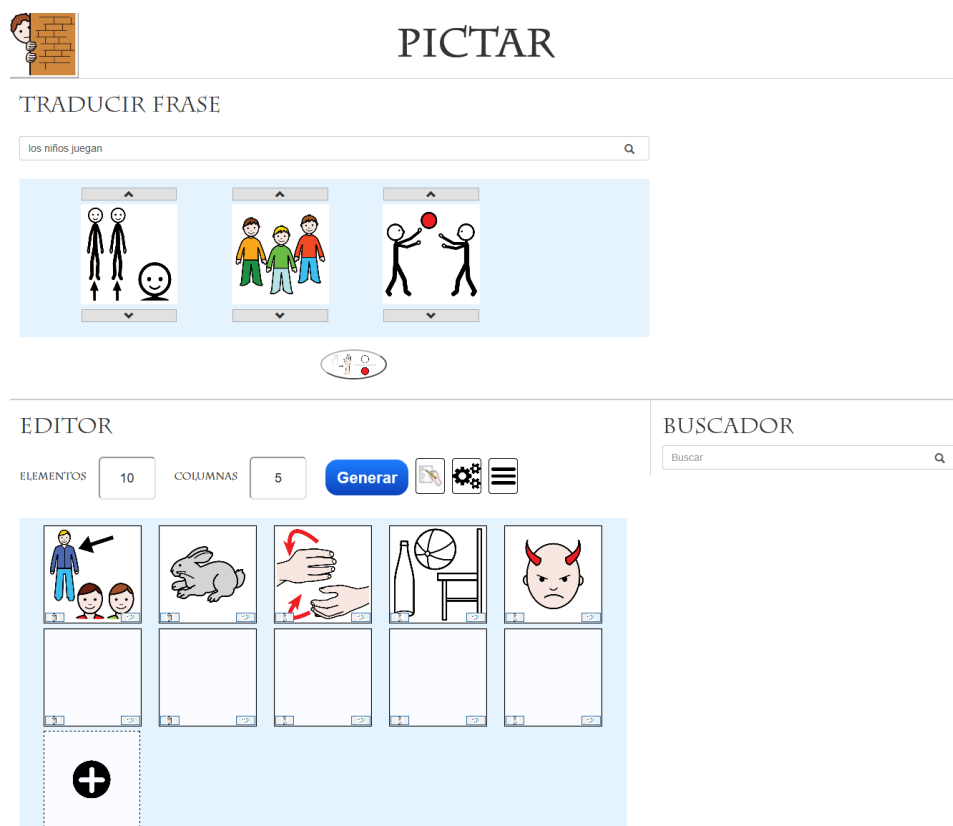


Figura 2.6: Pictar. <http://hypatia.fdi.ucm.es/pictar/>

**Pictotraductor** La idea de este proyecto es hacer más sencillo la comunicación con personas que presenten algún tipo de déficit en el lenguaje hablado. Es una aplicación web que permite traducir texto introducido en lenguaje natural a pictogramas (Ver figura 2.7)

En la parte central de la aplicación aparece un cuadro para introducir el texto que el usuario quiere traducir a pictogramas. Simultáneamente a la introducción del texto, se realiza la traducción. Si una palabra tiene asociados varios pictogramas aparecerá una flecha en la parte superior e inferior de cada imagen con la que se puede ver los distintos pictogramas disponibles y elegir el más adecuado.

Si un usuario se registra en la aplicación, se puede cambiar el tamaño de los pictogramas y del texto que los acompaña. Además, se puede modificar el color del borde o el fondo de la aplicación, escoger la posición del texto que aparece en los pictogramas, imprimirlos y compartirlos a través de correo electrónico o redes sociales. También se pueden guardar frases en favoritas y consultarlas en cualquier momento.

La aplicación también permite subir y utilizar pictogramas creados por

el usuario. Las imágenes subidas por éste, pueden ser empleadas para definir un término en lugar de los que ofrece la aplicación, o bien para añadir un pictograma para referirse a palabras que no sean tenidas en cuenta en la base de datos (pic, 2018b).

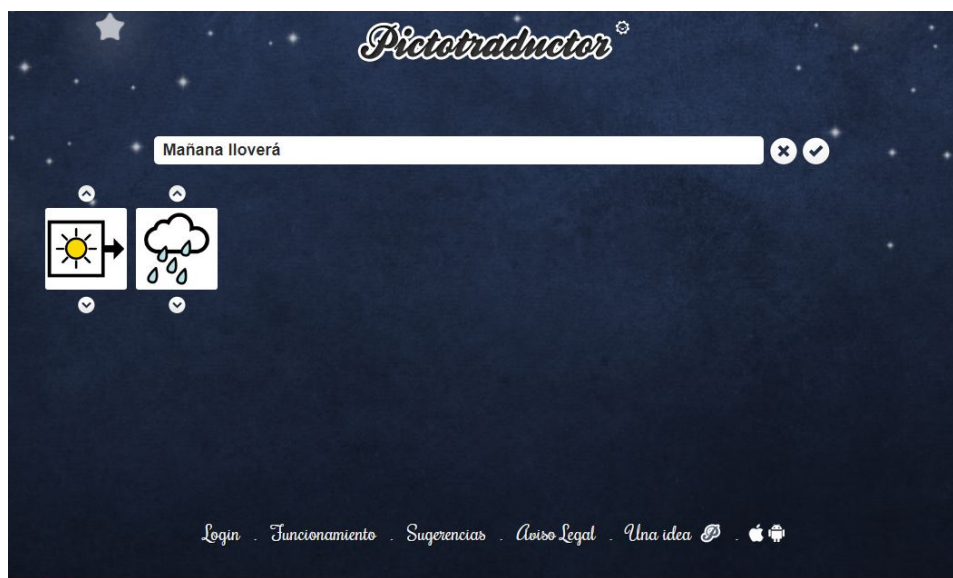


Figura 2.7: Pictotraductor. <https://www.pictotraductor.com/>

**Editor predictivo de mensajes en pictogramas** Es una aplicación creada por Carlos Ruiz Martín y Paloma Galván Calleja en 2014 para dispositivos móviles (Méndez et al., 2018).

Su principal funcionalidad es la implementación de un predictor que sugiere los siguientes pictogramas que se van a utilizar. La aplicación también ofrece la posibilidad de guardar y compartir composiciones de pictogramas.

Su diseño está adaptado al sistema operativo Android y utiliza la base de datos de pictogramas de ARASAAC.

La aplicación se divide en tres pantallas con las que el usuario puede interactuar. Su pantalla de inicio es bastante sencilla pues solo se muestran en ella varias imágenes con las distintas categorías de pictogramas.

La pantalla principal se divide en cuatro áreas: la que se encuentra más arriba simula una pizarra en la que se compondrá la oración. La segunda zona, en una pequeña definición de la categoría en la que se encuentra el usuario. La parte central, y la que más lugar ocupa en la

pantalla, contiene una imagen por cada categoría que se puede seleccionar. Debajo irán apareciendo los pictogramas que estén asociados a dicha categoría. La barra de colores que aparece a la izquierda de esta zona representa las categorías gramaticales, de esta forma se puede hacer un filtrado más preciso. La parte inferior de la pantalla presenta varios iconos a modo de botones, dos flechas con las que se puede cambiar el contenido, y dos botones con los que permutar entre los modos en los que se puede usar la aplicación.

La pantalla final presenta una imagen única con todos los pictogramas que quedaban como resultado en la pizarra de la pantalla principal. Además de la representación gráfica, también aparece la traducción a oración en texto. El usuario podrá guardar la frase en formato imagen en la galería del dispositivo Android, o compartirla a través de alguna aplicación instalada en el teléfono o tablet.

Como se ha dicho, la aplicación presenta dos modos de interacción, un modo de carga y otro de escritura que se explican a continuación:

- **Modo escritura:** es el que aparece por defecto cuando se abre la aplicación. En este modo, los pictogramas que se muestran para comenzar la estructuración de la frase, son aquellos que el usuario ha utilizado alguna vez o, si es la primera vez que se usa, los que vienen por defecto. En este modo se pueden borrar y recuperar pictogramas.



Figura 2.8: Editor predictivo de mensajes

- **Modo carga:** es el que tiene que utilizar el usuario para guardar pictogramas que quiera emplear a la hora de generar las oraciones. Tendrá la posibilidad de guardar hasta un máximo de treinta imágenes, que serán las que aparezcan en el modo escritura.

## 2.4. Predictores

En este apartado vamos a hablar de las formas más usadas en la actualidad para la predicción de texto, ya que nuestro proyecto utilizamos un algoritmo predictivo para obtener los pictogramas que es más probable que se vayan a utilizar después del que se acaba de introducir.

Actualmente, por ejemplo, los teclados de los móviles utilizan esta funcionalidad para que cuando una persona esté escribiendo le sugiera las palabras que suele utilizar más habitualmente después de las que ya tenga escritas. Nosotros lo vamos a utilizar para estudiar la probabilidad que tiene cada pictograma de salir después de otro, y así ofrecer al usuario una predicción adaptada.

### 2.4.1. Modelos de Markov

Deben su nombre al matemático ruso nacido en el siglo XIX Andrei Markov.

En el año 1913 Markov se preguntaba si sería posible aprovechar la frecuencia de aparición de las letras en el texto para ayudar a computar la probabilidad de que la siguiente letra del texto fuera una vocal.

El modelo oculto de Markov es un modelo de secuencia cuya labor es asignar una etiqueta o clase a cada unidad de la secuencia. Es un modelo probabilístico, que, dada una secuencia de palabras, letras, etc, calcula la probabilidad de la distribución de esta secuencia de etiquetas. El resultado es la devolución de la más probable.

Para definir correctamente el modelo de Markov primero debemos definir las **cadenas de markov**, también conocidas como modelo observado de Markov. Las cadenas y modelos de Markov son extensiones de autómatas finitos deterministas, ya que los autómatas están definidos como un conjunto de estados y transiciones entre esos estados. Con las transiciones se representa la probabilidad que tiene un estado de transitar a otro. Las cadenas de Markov son útiles para asignar probabilidades a secuencias de elementos (Marín Diazaraque, 2012).

Los modelos de Markov se utilizan por ejemplo en biomedicina para calcular el crecimiento de tumores, para predecir la estructura de las proteínas o modelar procesos biológicos, también se utilizan en el procesamiento de lenguaje, traducción automática (especialmente en reconocimiento de voz, de escritura manual o para generar etiquetas de categorías gramaticales), en

economía se utilizan para predecir la evolución de valores de bolsa. Estos modelos también se pueden utilizar para generar texto u organizar las páginas en internet, por ejemplo Google pagerank. (Jurafsky y Martin, 2017).

Dentro de los Modelos de Markov nos encontramos con varios tipos de implementaciones posibles.

La **Cadena de Markov** es un proceso probabilístico en el que la aparición de un evento solo depende del evento anterior, del que hablaremos posteriormente.

**Algoritmo de Viterbi.** Consiste en que dado un estado inicial se halle la secuencia más probable de siguientes estados teniendo en cuenta los estados anteriores.

**Algoritmo de Baum-Welch.** Consiste en encontrar un modelo para poder maximizar la probabilidad de que aparezca una secuencia de operaciones.

Después de estudiar todas las posibilidades decidimos en función de nuestras necesidades que algoritmo íbamos a utilizar.

El primer algoritmo que descartamos fue el algoritmo de Baum-welch ya que no se adaptaba a lo que nosotros necesitábamos.

Posteriormente estuvimos buscando información acerca del algoritmo de Viterbi para ver si era apropiado para resolver nuestro problema. Llegamos a la conclusión de que podría valernos en el caso de que necesitáramos los estados anteriores para realizar la predicción, pero no teníamos claro si necesitábamos la secuencia de estados anteriores o no.

Procedimos a buscar bibliotecas que implementaran este algoritmo y las principales que estudiamos fueron Mallet, jMarkov, jahmm en java y también estuvimos viendo si nos interesaba implementar este algoritmo en Matlab.

Las bibliotecas mencionadas anteriormente fueron descartadas debido a la falta de documentación y a que las modificaciones que se debían realizar eran demasiado significativas. Algunas implementaciones mas que se encontraron de este algoritmo estaban enfocadas a la creación de texto y la predicción de aminoácidos y bases nitrogenadas en las cadenas de ADN y en ambos casos teníamos el mismo problema que con las anteriores bibliotecas.

Con lo que llevábamos investigado llegamos a la conclusión de que no era necesario tener la información de los estados anteriores para predecir el siguiente ya que al contrario de las cadenas de ADN que cuentan con gran cantidad de estados, en nuestro caso las composiciones de pictogramas cuentan con un reducido número de estados y para predecir el siguiente estado nos sirve con el estado actual. Por lo tanto decidimos descartar el algoritmo de Viterbi y quedarnos con la Cadena de Markov, de la que vamos a hablar a continuación.



### 2.4.2. Cadena de Markov

Como hemos comentado anteriormente, la cadena de Markov se utiliza en procesos aleatorios para poder establecer una probabilidad de que se produzcan hechos en el futuro.

Para describir la Cadena debemos empezar por definir  $P$  como la probabilidad de que la cadena esté en el estado  $E$  después de  $n$  pasos. También debemos describir la matriz de transición  $T$  como una matriz de orden  $n$  en la que cada celda indica la posibilidad de transitar desde un estado  $i$  a un estado  $j$ , siendo  $T[i, j]$  la probabilidad de transitar del estado  $i$  al  $j$ .

Para calcular la probabilidad de transición de un estado a otro se debe calcular el producto del vector fila  $P$  de longitud  $n$ , por la matriz de transición  $T$  de orden  $n$  de modo que el vector  $P_1$  contiene la probabilidad de transición a cada estado en el siguiente paso. De este modo si quisiéramos calcular las probabilidades de transición para cada estado dentro de 2 pasos deberíamos calcular  $p_2$  como el producto de la matriz de transición por  $P_1$ , que indica la probabilidad de estar en cada estado en el paso 1.

La fórmula general para calcular las probabilidades de transición para el siguiente estado sería  $P_{n+1} = P_n \times T$  siendo  $P$  y  $T$  como se ve a continuación:

$$\mathcal{P} = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}$$

$$\mathcal{T} = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Para terminar de exponer la idea de la cadena de Markov a continuación se muestra un ejemplo de este concepto.

Supongamos que estamos utilizando un predictor basado en Modelos de markov para predecir la siguiente palabra. Tenemos las palabra a, b y c y las probabilidades de que se aparece la palabra a tiene un 75 % de probabilidades de que la siguiente sea a, un 15 % de que sea b y un 0.1 % de que sea c.

Si aparece b las probabilidades de que la siguiente palabra que aparezca sea a son de un 25 %, de un 60 % que sea b y de un 15 % que sea c.

Si la palabra que aparece es c, a tienen un 30 % de probabilidades de aparecer a continuación, b un 20 % y c un 50 %.

Suponemos que actualmente las probabilidades de que aparezca a es de un 60 %, b un 30 % y c un 10 %.

Vamos a calcular las probabilidades de que aparezcan a, b y c dentro de 2 palabras indicando la matriz de transición y la máquina de estados que indica las probabilidades de transición de cada estado.

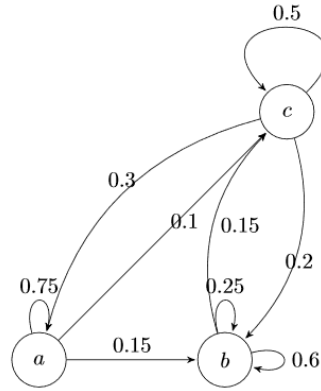


Figura 2.9: Probabilidades de transición de los estados

En la figura se muestran los estados que posee la cadena y las probabilidades de transición entre ellos (Ver figura 2.9).

La siguiente matriz  $T$  muestra las probabilidades de transición de cada estado y el vector  $P$  las probabilidades de estar en cada estado.

$$T = \begin{bmatrix} 0,75 & 0,15 & 0,1 \\ 0,25 & 0,6 & 0,15 \\ 0,3 & 0,2 & 0,5 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 0,6 & 0,3 & 0,1 \end{bmatrix}$$

Para calcular las probabilidades de aparición de la siguiente palabra debemos realizar la operación  $P_1 = P_0 T$  dando como resultado :

$$P_1 = \begin{bmatrix} 0,555 & 0,29 & 0,165 \end{bmatrix}$$

Este resultado indica que la probabilidad de que la siguiente palabra sea **a** es de un 55 %, **b** un 29 % y **c** un 16 %.

Ahora vamos a calcular la probabilidad de aparición de cada una de las palabras en el siguiente paso con la operación  $P_2 = P_1 T$  dando como resultado:

$$P_2 = \begin{bmatrix} 0,535 & 0,288 & 0,176 \end{bmatrix}$$

La probabilidad de aparición de la palabra **a** dentro de 2 pasos es de un 55 %, la de **b** un 29 % y la de **c** un 16 %.

### 2.4.3. Ngramas

Un n-grama es una sucesión de elementos (palabras, sílabas, pictogramas, etc) que unidos pueden tener un significado conjunto. Un n-grama puede representar uno o mas pictogramas.

Un modelo de n-grama es un modelo probabilístico que permite predecir el siguiente elemento de una secuencia dada. Los modelos de n-gramas se pueden definir mediante cadenas de markov de orden  $n-1$ .

Dado un conjunto de datos de aprendizaje se puede predecir la probabilidad de la aparición del siguiente elemento. Para un conjunto de tamaño  $n$  la suma de la probabilidad de cada elemento debe sumar 1.

Los n-gramas son útiles en la traducción de lenguaje natural ya que si una frase se tradujera elemento a elemento (unigramas) se perdería información respecto a si se traduce con trigramas o 5-gramas (n-gramas de 3 y 5 elementos). Por ejemplo, para la frase “ir de excursión” se podría agrupar en un trigramas en lugar de 3 unigramas.

Los n-gramas son ampliamente utilizados en informática, lingüística computacional y matemáticas. Se utilizan por ejemplo en algoritmos de aprendizaje automático para extraer información a través de cadenas de texto, reconocimiento de patrones para saber la probabilidad de que una palabra aparezca en un texto, predicción de texto, y en bioinformática se utilizan para analizar cadenas de ADN o aminoácidos. (Wikipedia, 2018a)

## 2.5. Diseño Centrado en el Usuario

El diseño centrado en el usuario se define como un enfoque de diseño dirigido por información acerca de las personas que van a hacer uso de la aplicación. En este tipo de diseño el usuario final prevalece sobre otros factores a la hora de tomar las decisiones, aunque eso no significa que los otros factores sean desatendidos (Hassan-Montero y Ortega-Santamaría, 2009).

En nuestro caso hemos elegido este tipo de diseño ya que vamos a realizar una aplicación dirigida a personas con TEA y vamos a realizar el diseño centrándonos en sus necesidades.

La experiencia de usuario no solo se centra en la funcionalidad, sino que también trata de abarcar aspectos que tengan en cuenta como se sienten los usuarios al interactuar con el sistema (Antonio Sánchez Ruiz-Granados, 2017).

Este tipo de diseños tiene en cuenta:

**Lenguaje del contenido:** tiene que ser claro y conciso, evitando los tecnicismos ya que pueden no ser comprensibles para el usuario.

**Diseño gráfico:** iconos sencillos y colores neutros que no desvíen la atención de los usuarios.

**El movimiento:** en nuestro caso, como nos estamos centrando en usuarios con TEA no puede haber muchos cambios puesto que afectan a la atención de los usuarios. En otro caso habría que estudiar al usuario y observar sus necesidades.

**Diseño de la información:** tiene que ser claro y seguir una estructura comprensible y sencilla.

**Diseño de la interfaz:** tiene que tener elementos entendibles y las interacciones deben ser comprensibles para los usuarios.

Los diseños centrados en el usuario tienen que hacer que este se sienta cómodo con el sistema y que haga que fluya la interacción, es decir, que el usuario se mantenga concentrado en lo que está realizando y que al mismo tiempo se sienta a gusto con ello. Para crear un diseño correcto hay que tener en cuenta que:

- Las tareas no son los objetivos. Hay que considerar cómo se realizan y quien las va a llevar acabo y cual es su objetivo.
- Los casos de uso deben tener en cuenta el facilitar la labor del usuario y tener una visión completa de los posibles usos, porque nunca va a ser lineal.
- No debemos basarnos en modelos formales, ya que estos tienden a centrarse en la interfaz y no en el usuario.

### 2.5.1. Usabilidad

Según la ISO la usabilidad es “Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones” (usa, 2019).

Esta característica se divide en la capacidad del producto para permitir al usuario entender si se adapta a sus necesidades, si el producto permite al usuario aprender su aplicación y operarlo y controlarlo con facilidad, proteger al usuario frente a hacer errores, capacidad de la interfaz de satisfacer la interacción con el usuario.

También hay que tener en cuenta el concepto de accesibilidad, que va ligado al de usabilidad, y que es la capacidad que permite al producto ser usado por usuarios con ciertas características y discapacidades.

Para tener un buen diseño interactivo y centrado en el usuario se deben seguir las siguientes fases:

1. **Conocer al usuario:** para ello hay que realizar entrevistas con los usuarios y documentarse sobre ellos.

Este paso se denomina modelado del usuario y consiste en realizar perfiles de usuarios en base a atributos que tienen en común. Estos atributos dependen de la información de la que se disponga pero normalmente serán necesidades de información, condiciones de acceso, experiencia y conocimientos.

Gracias a esta técnica el diseñador tiene en cuenta para quién está diseñando, qué espera encontrarse el usuario y la forma en la que espera encontrárselo.

Los sitios web deben estar diseñados en función del usuario, organizando y estructurando la información según los modelos de usuario.

**2. Análisis de la competencia:** Se buscan aplicaciones similares a la que se quiere implementar, se extrae información sobre el diseño, la forma de expresar la información o la interacción con el usuario de las interfaces ya creadas buscando fallos para poder eliminarlos de nuestra aplicación.

A continuación se realiza el diseño de la interfaz propia pudiendo tomar ideas del análisis realizado.

**3. Fijar objetivos:** tras los dos pasos anteriores, ya se tiene una idea de lo que se busca y con ello se comienzan a realizar los objetivos de diseño concreto que se quieren conseguir.

**4. Realizar diseños:** una vez claros los objetivos, se comienza a crear bocetos en los que basarse.

**5. Prototipado:** a partir de los bocetos, se comienza a trabajar en un prototipo que sea más visual, y con el que tener una idea del funcionamiento del sistema.

Al principio se realizan prototipos de baja fidelidad, normalmente en papel ya que son fáciles de realizar, y se pueden cambiar rápidamente.

Más adelante se realizan prototipos de alta fidelidad, con un alto nivel de detalle.

**6. Evaluación:** una vez obtenido un prototipo válido se realiza la evaluación por los usuarios o por expertos en el tema en el que se desarrolla la aplicación para ver los aspectos positivos y negativos.

La evaluación se puede realizar por inspección con eucarísticas o mediante test con usuarios.

**7. Seguimiento:** Dado que un sitio web no es estático los contenidos pueden cambiar por lo que requiere seguimiento, además de rediseños y mejoras.

El seguimiento se puede realizar mediante las opiniones de los usuarios, mediante formularios o encuestas. También se puede realizar mediante el comportamiento de los usuarios o del sitio web.

#### **2.5.1.1. Conocer al usuario**

Esta fase consiste en el análisis de los usuarios que vayan a hacer uso de la aplicación. Los diseñadores son los que tienen que entrevistarse con los usuarios finales y no terceros ya que son los que tiene que obtener la información sobre los usuarios para a continuación realizar perfiles sobre ellos. Tres son las preguntas que hay que hacerse durante esta fase:

- ¿Qué usuarios van a emplear este sistema?
- ¿De que forma se va a utilizar el sistema?
- ¿Qué objetivo se busca alcanzar?

Durante este análisis hay que comprender que tareas va a llevar a cabo el usuario y para ello hay que observarlos durante la realización de la misma. Además de observar a los usuarios, sería bueno realizar un análisis funcional de forma que estos puedan expresar sus necesidades.

#### **2.5.1.2. Análisis de la competencia**

El análisis de la competencia consiste en analizar la forma en que los usuarios utilizan un producto existente para comprender cómo interactúa el usuario con el sistema y que problemas de usabilidad existen para poder evitarlos en el sistema que vamos a diseñar.

En esta fase se pretende obtener información acerca de las características que resultan útiles o atractivas a los usuarios y los problemas que estos se encuentran.

El proceso de análisis se puede realizar en cuatro etapas:

- En la primera etapa se identifica la competencia, creando una lista con aplicaciones que ofrecen una funcionalidad similar a la que queremos implementar.
- En la segunda etapa nos centramos en las funcionalidades que nos parecen mas relevantes. La búsqueda hay que centrarla en la perspectiva del usuario. Esto sirve para limitar la búsqueda y en nuestro caso nos hemos centrado en aplicaciones que hacen uso de pictogramas y disponen de funcionalidades para organizar estos pictogramas o programar tareas con ellos.

- En la tercera etapa nos centramos en la competencia que hemos identificado y sobre esta competencia observamos que funcionalidades cubre cada aplicación.
- En la última fase se obtienen las cosas positivas que implementa la competencia y los defectos, y con esto podríamos hacernos una idea sobre las funcionalidades que crean una buena experiencia de usuario, los problemas que hay en la competencia para evitar cometerlos y como poder mejorar lo que ya existe.

A continuación explicamos brevemente la competencia que hemos encontrado ya que en el capítulo Estado de la Cuestión se explican con mas detalle cada una de las aplicaciones anteriores (ver Apartado 2.3.2).

**Pictoagenda** Es una aplicación web creada para diseñar actividades utilizando pictogramas.

**PICTAR** Es una aplicación creada por Alejandro Martín que permite traducir texto a pictogramas, además de guardar las composiciones de pictogramas.

**Pictotraductor** Es una aplicación web que permite traducir lenguaje natural a pictogramas.

**Editor predictivo de mensajes en pictogramas** Esta aplicación incorpora la funcionalidad de sugerir los siguientes pictogramas que se van a utilizar.

Todas las aplicaciones anteriores utilizan los pictogramas de ARASAAC.

### 2.5.1.3. Fijar objetivos

Al comienzo del proyecto nos realizábamos las siguientes preguntas: ¿Cuál va a ser el tipo de usuario? ¿Qué funcionalidades va a proporcionar? ¿Cuál es el contexto de uso?

A partir de estas preguntas investigamos acerca del tipo de usuario que va a utilizar nuestra aplicación y aplicaciones similares, como ya hemos visto en el apartado anterior.

Después de investigar hemos descubierto las necesidades de las personas con TEA y el tipo de aplicaciones en las que podemos basarnos para poder desarrollar la nuestra (Paz y Szyszlican, 2014).

### 2.5.1.4. Prototipado

En esta etapa se elaboran prototipos de la interfaz para la evaluación del sitio web. Los prototipos no tiene por qué ser exactos respecto a lo que sería la interfaz una vez el trabajo haya finalizado, pero pueden ser de utilidad para evaluar la usabilidad de lo que se quiere implementar antes de implementarlo.

Los prototipos se pueden clasificar de acuerdo a la funcionalidad que se reproduce.

**Prototipado horizontal.** Se reproduce el aspecto visual pero sin incorporar la funcionalidad real que tendría una vez implementado.

**Prototipado vertical.** Se reproduce el aspecto visual con la misma funcionalidad que tendría la parte que se reproduce una vez se haya implementado.

Los prototipos también se pueden clasificar en función de la calidad en la que se reproducen.

**Prototipado de baja fidelidad.** El prototipo no es muy parecido al aspecto final que tendría el sitio web.

**Prototipado de alta fidelidad.** Los prototipos tienen un aspecto muy similar al que tendría el sitio una vez se haya terminado.

En las primeras etapas de desarrollo se suelen utilizar prototipos de baja fidelidad como los realizados en papel, ya que se pueden realizar rápidamente y con ellos se pretende dar una primera impresión sin muchos detalles del aspecto final.

Después se pueden realizar prototipos de alta fidelidad con herramientas de diseño como **Just in Mind** o **HTML** para realizar una aproximación mas detallada sobre el aspecto final.

#### 2.5.1.5. Evaluación

La forma mas habitual de llevar a cabo la evaluación de la usabilidad es la **evaluación heurística**. Esta inspección se basa en realizar un recorrido para encontrar errores y problemas de diseño. Su principal ventaja es que se puede llevar a cabo con gran rapidez.

Esta evaluación se suele llevar a cabo por un grupo de evaluadores reducido, que se basan en su experiencia, principios de usabilidad heurísticos y apoyándose en unas guías que se elaboran con la intención de evaluar cada caso de forma individual e independiente.

Finalmente se ponen los resultados de todos los evaluadores en común.

Dependiendo de cada autor se han propuesto diferentes heurísticas. Nosotros vamos a fijarnos en las de Nielsen que ya hemos utilizado en la asignatura Diseño de Sistemas Interactivos, las cuales vamos a describir brevemente a continuación (Nielsen, 2005).

- **1. Visibilidad del estado del sistema.** El sistema siempre tiene que informar al usuario acerca de lo que está sucediendo.
- **2. Lenguaje común entre sistema y usuario.** El sistema debe hablar el mismo lenguaje que el usuario, es decir, debe evitar usar tecnicismos que el usuario no comprende.
- **3. Libertad y control por parte del usuario** El usuario debe controlar el sistema evitando limitar su actuación.



- **4. Consistencia y estándares.** Intentar seguir estándares de diseño que están ampliamente aceptados. Mientras mas se parezca el diseño y funcionamiento al resto de sitios web mas fácil resultará su uso para el usuario.
- **5. Prevención de errores.** Es mejor tener un diseño que evite que ocurra un error a tener un buen mensaje de error.
- **6. Es mejor reconocer que recordar.** Este principio se refiere a la visibilidad de las opciones ya que el el usuario no tiene por qué recordar donde se encontraba la información o como llegar a un determinado sitio.
- **7. Flexibilidad y eficiencia de uso.** Debe estar diseñado para que sea fácil de usar para los usuarios novatos y proporcione atajos a los usuarios avanzados.
- **8. Diseño minimalista.** Hay que eliminar la información que no sea relevante para evitar sobrecargar la interfaz.
- **9. Permitir al usuario solucionar el error.** Se debe informar al usuario como se puede solucionar un error por ejemplo con mensajes de texto.
- **10. Ayuda y documentación.** Es mejor poder utilizar un sitio sin necesidad de ayuda o de leer la documentación, aunque si se trata de un proceso complejo se proporciona ayuda sobre la usabilidad al usuario.

#### 2.5.1.6. Estudios de seguimiento

El trabajo de diseño de una aplicación web no termina con la implementación y entrega al cliente, ya que las necesidades y contenidos cambian. Por lo tanto requiere trabajos de mantenimiento y seguimiento.

Los cambios realizados no se pueden realizar de forma drástica o incontralada, sino que los principales cambios se realizan para solucionar problemas de usabilidad. Estos cambios hay que realizarlos con mucho cuidado ya que los usuarios ya estaban acostumbrados al diseño actual.

Los problemas que no se han detectado durante el proceso de desarrollo se pueden descubrir principalmente a través de dos métodos, los cuales vamos a explicar a continuación.

- **Opiniones de los usuarios.** Estas opiniones se pueden obtener de forma pasiva como por ejemplo por medio de mensajes de los usuarios, o de forma activa mediante la realización de cuestionarios o encuestas. Hay que tener en cuenta que estas respuestas sirven para indicar problemas pero no son soluciones a los problemas detectados. Las respuestas

de las encuestas no sirven para evaluar la usabilidad del sitio, sino que sirven para evaluar la satisfacción del usuario.

- **Comportamiento del usuario y uso del sitio.** Cuando el sitio web ya está funcionando se puede adquirir información del comportamiento de los usuarios mediante ficheros de `'log'`. Estos ficheros son generados por el servidor y contienen información como la IP del cliente, la identidad de los usuarios o fecha y hora de las peticiones al servidor.

Mediante el análisis de estos ficheros se pueden conocer datos de uso de los usuarios y sus hábitos como la hora o páginas visitadas.

Esta información es importante a la hora de tomar decisiones para realizar rediseños.

## 2.6. Conclusiones

En este capítulo hemos investigado acerca de los usuarios a los que va dirigida la aplicación y sus necesidades, también hemos realizado un análisis de las aplicaciones que hacen uso de pictogramas para conocer lo que ya existe. Después investigamos acerca de las formas que se utilizan actualmente para realizar predicciones y eligiendo la que creemos que es mejor para resolver nuestro problema, y por último estuvimos estudiando el Diseño Centrado en el Usuario ya que pensamos que es la metodología de diseño que debemos utilizar.

Con los conocimientos adquiridos en este capítulo acerca de los usuarios y las tecnologías que existen actualmente ya podemos pasar a continuación a realizar el diseño de la interfaz gráfica de nuestra aplicación.

# Capítulo 3

## Diseño de la interfaz

*“Nobody dies a virgin. Life fucks us all.”*  
— Kurt Cobain

En este capítulo vamos a realizar el desarrollo de la interfaz de nuestra aplicación basándonos en otras que hemos estudiado en el capítulo del estado de la cuestión. Además vamos a tener en cuenta las necesidades de personas con TEA. Al diseñar nuestra aplicación hemos seguido el diseño centrado en el usuario visto en el capítulo anterior.

### 3.1. Diseño competitivo

El diseño competitivo consiste en buscar una solución a un problema dado de forma individual y sin observar o tener información del trabajo realizado por los otros miembros del equipo.

Una vez se han encontrado las soluciones, los miembros que participan en el diseño las exponen, y una vez finalizas todas las exposiciones, se escogen las mejores ideas, se descartan las menos válidas e incluso se puede aceptar una única idea. Los miembros del equipo son los encargados de ponerse de acuerdo a la hora de elegir y descartar los diseños propuestos.

Cuando ya se tienen las ideas con las que se quiere trabajar, los miembros del equipo trabajan en conjunto para comenzar a diseñar una solución al problema.

Hemos escogido seguir este tipo de diseño para realizar nuestra aplicación porque nos parece que es la mejor forma de que cada miembro de nuestro equipo exponga sus ideas de diseño libremente y sin tener la influencia de otros miembros del equipo a la hora de plantear su solución, y a partir de ahí poder ponerlo en común y refinar el diseño hasta llegar a la versión final con las mejores ideas.

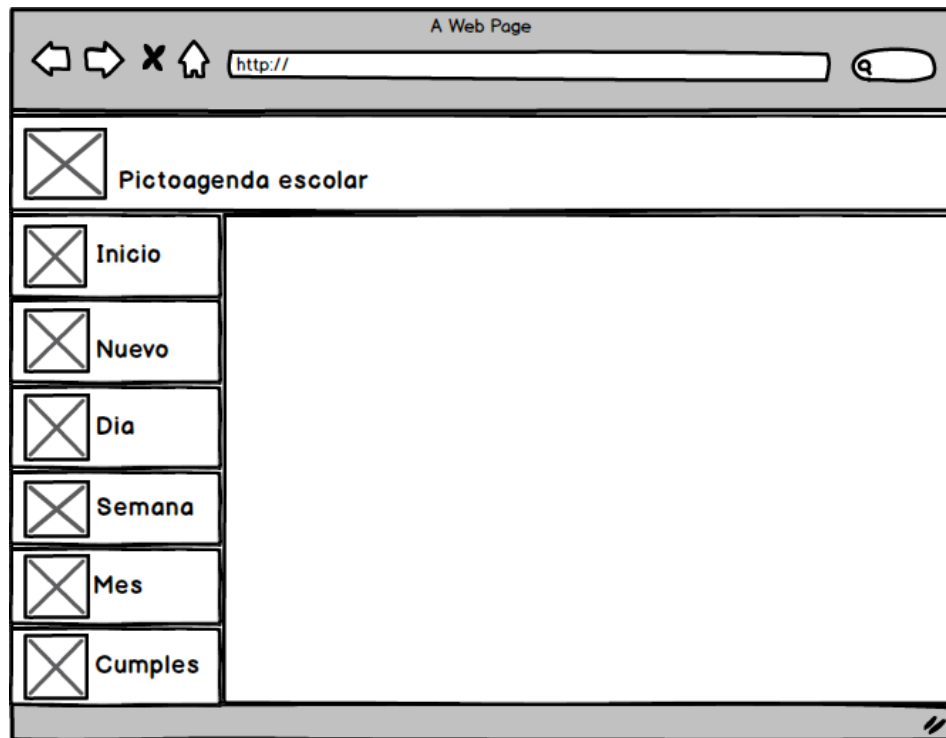


Figura 3.1: Vista diseño inicial de Mónica

## 3.2. Primera versión

En esta primera versión hemos realizado el diseño de la interfaz por separado como hemos explicado en el punto anterior.

Para desarrollar esta versión hemos tenido en cuenta las necesidades de las personas con TEA.

Los recursos deben ser simples, intuitivos y fáciles de entender, además de predecibles y adaptables.

### 3.2.1. Versión de Mónica

La parte superior de la ventana es la cabecera, con el logotipo, el título y donde irán los enlaces al registro o al login de la aplicación.

En la parte izquierda de la ventana aparece un menú que siempre está fijo y que muestra al usuario varias opciones para acceder a ellas de forma sencilla y rápida (ver Figura 3.1).

**Inicio:** será la pantalla principal por defecto en la que se explica a que está dedicada la aplicación, etc.

**Nuevo:** será donde los niños/as, padres, madres y/o profesores/as, creen las

nuevas entradas en la agenda.

**Día, semana y mes:** está pensado para tener una visión rápida de los calendarios. De esta forma, si se quiere ver la entrada específica de un día, se puede acceder a él sin ver todo lo demás.

**Cumples:** se ha creado esta pestaña porque a todos los niños les gusta llevar en sus agendas los cumpleaños de sus compañeros para poder felicitarles cuando llegan a clase, al igual que les gusta que les feliciten a ellos.

### 3.2.2. Versión de Iván

A la hora de diseñar la interfaz he tenido en cuenta que los usuarios que utilizan pictogramas no escriben. Por lo tanto la mayor parte del espacio está pensado para interactuar con pictogramas y no con texto, pero se deja un espacio para poder introducir texto y traducirlo a pictogramas, para dar la posibilidad de comunicarse con usuarios que solo utilizan los pictogramas.

En la parte superior derecha de la pantalla hay un cuadro para introducir texto y convertirlo a pictogramas.

Debajo aparece el mensaje traducido a pictogramas que cambiará de forma dinámica según se vayan introduciendo estos al mensaje.

Mas abajo, aparecen las sugerencias de pictogramas obtenidos en función del usuario. Está pensado que no aparezcan muchos pictogramas y cambien lo menos posible para no confundir al usuario, además que de este modo se pueden con un tamaño mayor para facilitar su visualización.

En la parte izquierda aparecen las categorías de pictogramas que seleccionando una de ellas en la parte derecha de la pantalla aparecen los pictogramas de esa categoría. Al seleccionar un pictograma aparecerá en el mensaje al igual que si se introdujera por texto.

Se podrán añadir y eliminar categorías además de poder agregar pictogramas personalizados.

En la parte inferior de la ventana aparecerán las composiciones de pictogramas que se han realizado en el mensaje (ver Figura 3.2).

## 3.3. Segunda versión

A partir de esta segunda versión el diseño está realizado conjuntamente por ambos miembros, poniendo en común las ideas y basándose en los prototipos realizados en el punto anterior.

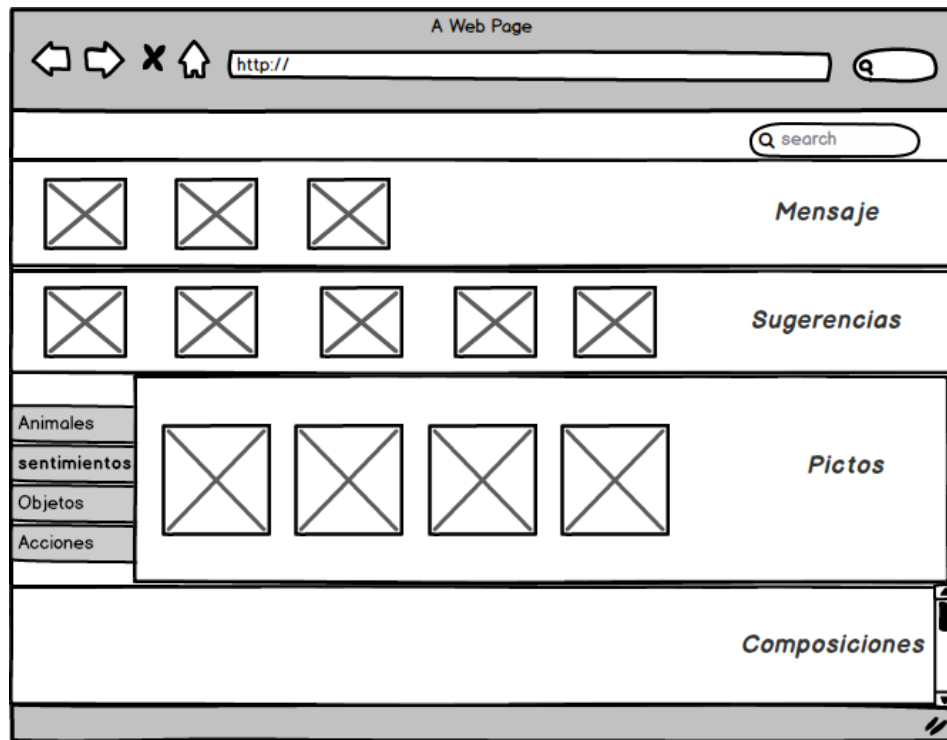


Figura 3.2: Vista diseño inicial de Iván

### 3.3.1. Escoger ideas de ambas versiones

En este punto ponemos ambos diseños en común y escogemos las ideas que mejor nos han parecido para desarrollarlas y continuar desarrollando una versión conjuntamente.

De la versión de Mónica hemos escogido el botón de **login** y la idea de un menú.

De la versión de Iván hemos seleccionado la vista del mensaje y las sugerencias además de la idea de introducir texto.

Esta última idea se ha modificado incluyendo dos botones para poder introducir texto o pictogramas en el mensaje según el botón que se seleccione. Se ha hecho así para intentar que se produzcan los menores cambios posibles al usar la interfaz. Otra idea que se planteó fue crear dos vistas separadas, una para introducir texto y otra pictogramas pero nos parecía mejor la anterior idea.

De ambos diseños se ha obtenido la idea de agrupar los pictogramas por categorías, pudiendo añadir y quitar categorías

### 3.3.2. Página de inicio

La parte superior de la pantalla representa la cabecera de la aplicación. En la esquina izquierda aparece un símbolo que representa un menú desplegable. Lo hemos hecho así para que no recargue la página y solo se utilice cuando sea necesario y así poder navegar por las diferentes páginas de la aplicación web. En la parte central de la cabecera aparece el nombre de la aplicación y en esquina derecha, aparecen las opciones para iniciar sesión y para registrarse.

La parte central de la pantalla es la parte principal. Por lo tanto, es la que se irá modificando según se navegue por las distintas páginas de la aplicación. En la parte de arriba aparecen las opciones para introducir datos, ya que los usuarios podrán insertar la información en forma de texto o de pictogramas.

Más abajo se muestra el mensaje que se está componiendo mediante la traducción del texto introducido o con los pictogramas que se vayan seleccionando.

Debajo del mensaje aparecen las categorías de pictogramas que se pueden seleccionar. En la inferior están las sugerencias de las categorías seleccionadas. Si no hay ninguna categoría, aparecerán como sugerencia, los pictogramas que el usuario registrado haya utilizado con mayor frecuencia (ver Figura 3.3).

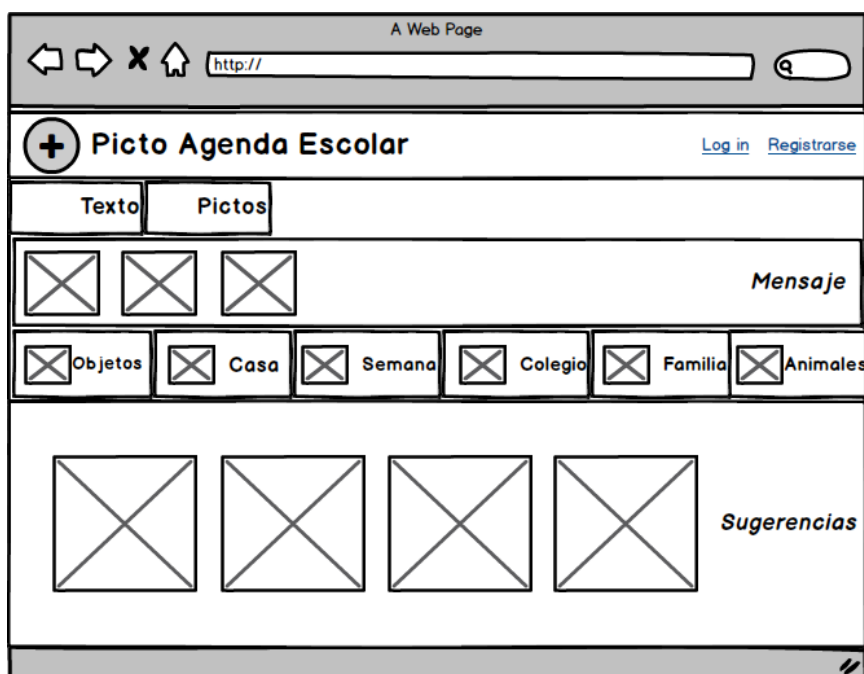


Figura 3.3: Vista página de inicio versión 3

### 3.3.3. Menú desplegable

Al pulsar en el icono del menú de la parte superior derecha aparece el menú desplegable con las opciones **Inicio**, **Composiciones**, **Calendario** y **Cumples**.

Al seleccionar la pestaña **Calendario** aparecen las opciones **Día**, **Semana** y **Mes** (ver Figura 3.4).

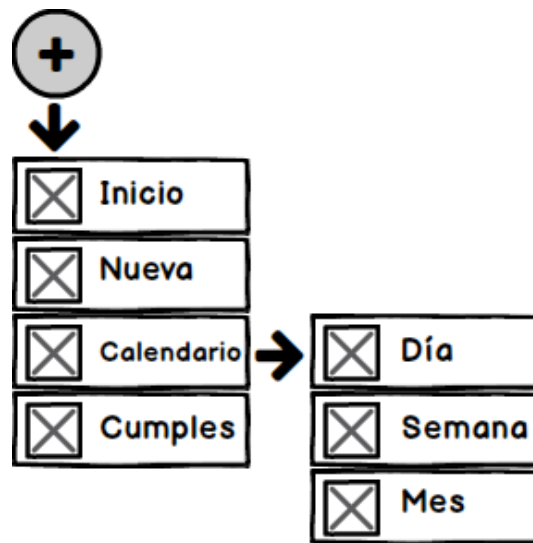


Figura 3.4: Vista menú versión 3

### 3.3.4. Inicio

Al selecciona esta opción en el menú desplegable regresa a la pantalla principal desde cualquier otra pantalla en la que se encuentre el usuario.

Si ya se encuentra en la pantalla de inicio no hace nada.

### 3.3.5. Composiciones

En esta pantalla aparecen las composiciones de pictogramas guardadas que se han realizado en la pantalla principal de la aplicación.

La parte superior de la pantalla es igual al resto de opciones de la aplicación para intentar que cambien los menores elementos posibles cuando se cambia de una opción a otra.

En cada línea se muestra una composición diferente.

En la parte derecha de la pantalla aparece una barra para hacer scroll y poder ver todas las composiciones guardadas.

Teníamos otra opción que era ubicar las composiciones en la pantalla principal pero pensamos que la opción elegida es mejor ya que por lo estu-



diado en el Capítulo 2 las personas con TEA se adaptan mejor y les es más sencillo asimilar la información si se producen los menores cambios posibles. Con la otra opción nos parecía que tener que hacer scroll para ver las composiciones y luego volver al editor es menos intuitivo que dar a un botón (ver Figura 3.5).

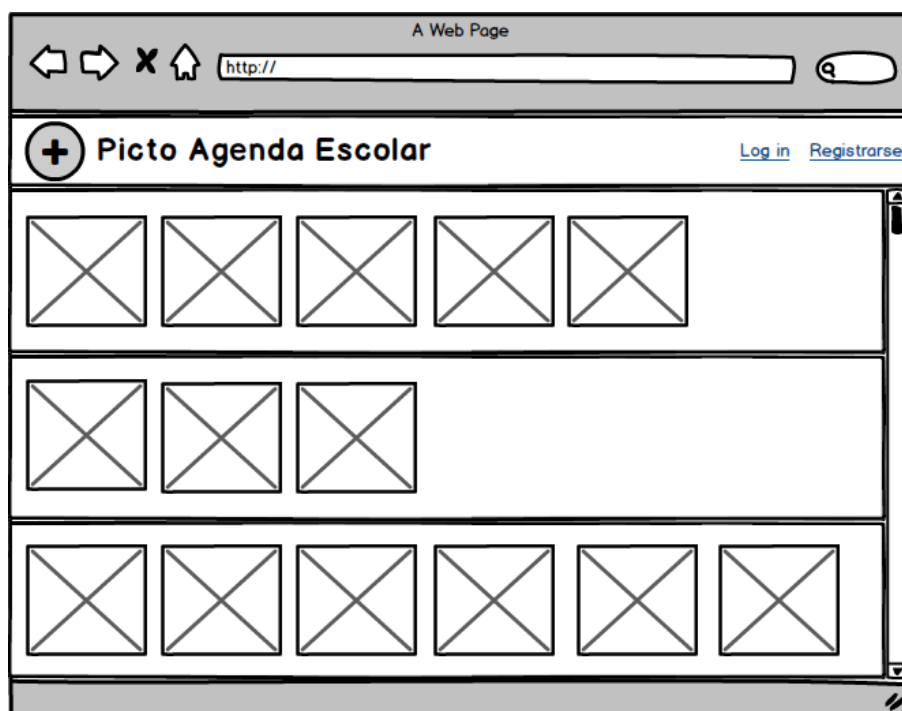


Figura 3.5: Vista página Composiciones versión 3

### 3.3.6. Calendario

En esta pantalla aparecen las tareas y actividades programadas en el día, la semana o el mes en función de la opción que se seleccione en el menú.

Se podrán añadir y eliminar tareas.

También se podrán ver eventos pasados o anotaciones que haya hecho cada persona (ver Figura 3.6).

### 3.3.7. Cumpleaños

En esta pantalla aparecen la relación de usuario con la persona que cumple años, junto con la fecha, la persona que cumple años y la fecha del cumpleaños.

Se podrá añadir y eliminar cumpleaños de la aplicación (ver Figura 3.7).



Figura 3.6: Vista calendario versión 3

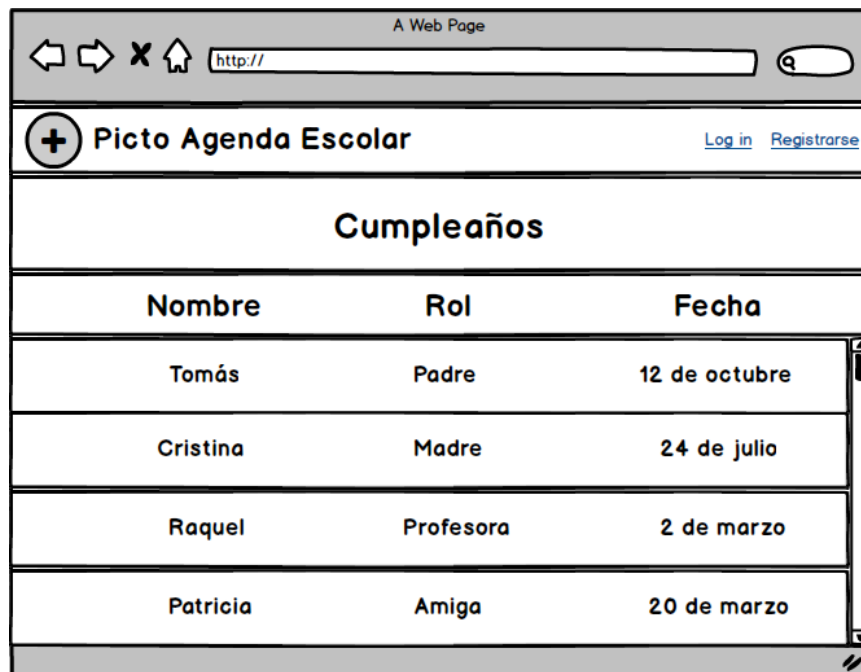


Figura 3.7: Vista página cumpleaños versión 3

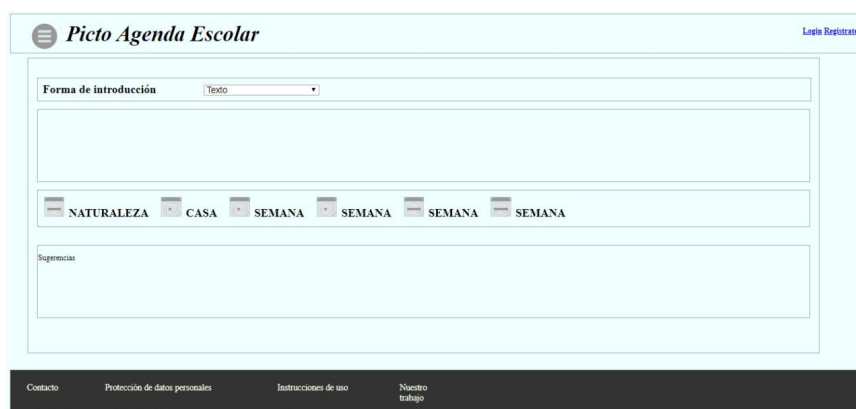


Figura 3.8: Vista de la primera versión implementada

### 3.4. Primera versión implementada

La primera versión que se diseña es una mezcla de los bocetos explicados anteriormente. El diseño de la interfaz es sencillo en colores claros tendiendo a que sean neutros para evitar posibles distracciones con colores llamativos. La cabecera está compuesta por un menú desplegable con varias opciones, el nombre de la página y los links para inicio y fin de sesión. Como puede verse, los links quedan poco visibles debido a su tamaño. Esto es algo que se busca mejorar con la siguiente versión. La pantalla principal está formada por tres partes. En primer lugar se encuentra un cuadro, que según que opción se seleccione en el dropdown (desplegable), habilita el modo de introducción de las frases. A continuación, se encuentra una barra con las categorías en las que se encuentran divididos los pictogramas. Y, por último, al final de la pantalla se encuentra un nuevo espacio, el cual está diseñado para las sugerencias que vayan apareciendo a la hora de redactar las oraciones. Esta parte estaría reservada a las predicciones. No es un diseño muy cómodo, ya que las sugerencias están bastante separadas de la zona de escritura y puede resultar algo incómodo para los usuarios. Éste es uno de los motivos por lo que se decide crear un nuevo diseño más adaptado y sencillo. En la parte posterior de la página se encuentra el pie de página. Está será la parte menos modificada de la página, puesto que solo contiene enlaces informativos, que se mantendrán en el nuevo diseño. (Ver figura 3.8).

### 3.5. Versión final

En esta última versión se ha refinado el diseño respecto a la versión anterior, eliminando las partes que creíamos que podrían ser menos útiles o que menos se utilizarían. Un ejemplo de estos cambios es el menú de la cabecera, que como se puede ver, ha desaparecido por completo. La página

está pensada sobre todo para niños y su comunicación por lo que algunas opciones del menú han sido eliminadas.

Una de las ideas principales de esta página es que pudiese utilizarse como agenda, por ello se encontraban las opciones de los calendarios en el menú. Este diseño se ha transformado y ha pasado a estar en el perfil de usuario. Esta decisión se ha hecho para que los usuarios no pierdan la funcionalidad de agenda, y así puedan mantener frases guardadas según la fecha pero manteniendo un diseño principal sencillo y limpio, sin obstáculos para la vista.

Los links de inicio de sesión se han convertido en dos botones, de diseño sencillo pero con mayor visibilidad que los enlaces que le precedían. Y se convirtieron en el de perfil y el cierre de sesión, de forma que no modifiquen su diseño.

El título se ha acortado, poniendo nombre propio, y como imagen se ha decidido el puzzle de cuatro colores ya que simboliza a los niños con TEA(Síndrome del Espectro Autista).

El pie de página ha mantenido la mayor parte de su diseño, puesto, que como se ha explicado anteriormente, solo son enlaces a páginas informativas.

El gran cambio lo ha sufrido la pantalla principal, que es el foco principal del usuario. Ha sido un cambio brusco pero se buscaba una mejor visibilidad de todas aquellas áreas que la forman. Aparte, los colores se han modificado y se han buscado líneas menos llamativas que permitan una mejor atención a los usuarios. En el diseño anterior la página se dividía en tres áreas, que ahora han pasado a ser cuatro, ya que se ha añadido un buscador de pictogramas. La parte de categorías y sugerencias se han unificado en una sola zona que aparece a la derecha, al igual que el buscador. La decisión de colocarlo en la parte derecha de la pantalla se debe a que esa zona capta menos la atención del usuario, por ello, al ser más cambiante por la variedad de imágenes que pueden aparecer, queda desplazada a la derecha.

A continuación se describirá más detenidamente cada parte de la página.

### 3.5.1. Página principal

Es la pantalla donde más tiempo permanecerá el usuario ya que es donde se realizan tanto las traducciones como las composiciones con pictogramas. Para una mejor visualización y comodidad a la hora de utilizarla se ha dividido en cuatro zonas:

En la parte superior izquierda aparece el icono con una casa dibujada que sirve para moverte a la página principal de forma rápida. En la parte central aparecen el logo y el nombre de la aplicación. Mientras que en la parte derecha aparecen los botones para iniciar sesión y registrarse (Ver figura 3.9).

Si se presiona el botón **Regístrate** se muestra un formulario de registro en el que se piden los datos del usuario para añadirlo a la base de datos y así

tenga acceso a más funcionalidades que se explicarán en la parte del perfil.

Si se presiona el botón **Inicia Sesión** se pide el email y la contraseña del usuario para iniciar sesión.

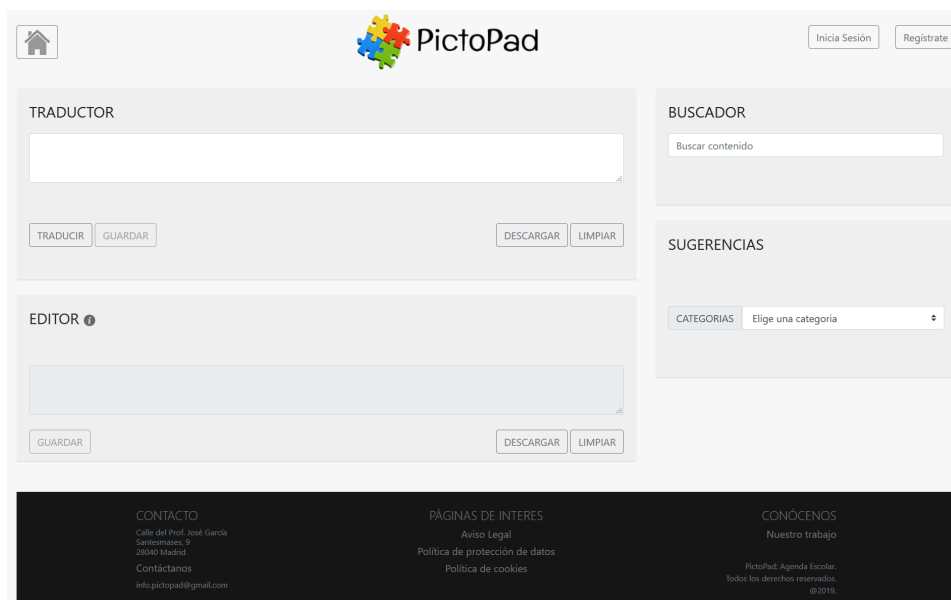


Figura 3.9: Vista de la página principal de la versión final

Dentro de la parte central de la aplicación en la parte izquierda aparece el **Traductor**. Para utilizarlo solo hay que introducir un texto en lenguaje natural y de forma inmediata aparece traducido más abajo en forma de pictogramas. Como opciones se encuentra guardar la traducción para que en cualquier momento se pueda acceder a ella desde el perfil de usuario. También se encuentra la opción de limpiar, que lo que hace es reestablecer toda la parte del traductor eliminando tanto el texto que hemos introducido como los pictogramas que lo representaban. Y por último está la opción de descargar, que genera un documento en formato PDF con la frase tanto en modo texto como en pictogramas, y en la siguiente hoja se genera una explicación de las distintas imágenes que se han generado al traducir el texto.

Debajo del **Traductor** se encuentra el **Editor**. Este espacio está creado para que se puedan componer oraciones a partir de pictogramas. Estos son elegidos usando el **buscador** o la parte de **sugerencias**. Debajo del grid o zona donde aparecen las imágenes, se encuentra un cuadro sombreado ya que está en modo lectura. Ahí es donde, según se vayan eligiendo los dibujos, irá apareciendo la respectiva traducción a texto natural. Al igual que el predictor, cuenta con las opciones de guardar, descargar y limpiar las composiciones.

En la parte superior derecha aparece el **Buscador**. Va haciendo una búsqueda continua que coincida con el texto que se está buscando, es decir, cada

vez que se va añadiendo una nueva letra para formar la palabra que se quiere encontrar, el buscador hace una consulta a base de datos para mostrar las imágenes que mayor coincidencia tengan con dicha palabra. En el caso de no haber coincidencias aparece una alerta informando de que no ha habido resultados.

Debajo, en la parte inferior derecha, se encuentran las **Sugerencias**. El espacio entre el título y las categorías está reservado para las sugerencias que salgan del predictor. En las categorías se muestran los pictogramas que estén relacionados con ella. Al seleccionarlos se mostraran en el **Editor**.

En la parte inferior, al igual que en la primera versión implementada se muestra información relevante de la página.

### 3.5.2. Mi Perfil

En la parte superior de la pantalla y en la parte inferior no se produce ningún cambio con respecto a la pantalla principal (Ver figura 3.10).

The screenshot shows the 'Mi Perfil' page of the PictoPad application. The header includes a home icon, the PictoPad logo, and buttons for 'Mi Perfil' and 'Cerrar Sesión'. The main content area is divided into three columns. The left column contains 'Datos Personales' (Name: Iván, Surname: García, Email: ivagar04@ucm.es, Birth Date: 04 de Agosto de 1994, and a 'Dar de baja' button), 'Cambio de contraseña' (Current, New, and Repeat password fields with a 'Cambiar Contraseña' button), and 'Mis frases' (a date picker). The middle column contains 'Pictogramas actuales' and 'Nuevo pictograma' (with an 'Examinar...' button and a 'Subir imagen' button). The right column contains a footer with 'CONTACTO', 'PÁGINAS DE INTERÉS', and 'CONÓCEMOS' sections.

Figura 3.10: Vista de la página Mi Perfil de la versión final

En la parte izquierda aparecen los **Datos Personales** del usuario junto con el email y un botón para poder darse de baja de la aplicación. Debajo, se puede realizar un **Cambio de Contraseña**

En la parte derecha aparecen los pictogramas propios del usuario y justo debajo se da la opción de subir nuevas imágenes para crear pictogramas personalizados para el usuario.

En la parte inferior de la página aparecen **Mis Frases**, donde se muestran las frases que ha creado y guardado el usuario. Por defecto aparecen las más recientes, pudiéndose filtrar por la fecha en la que fueron creadas.

### 3.5.3. Panel de administración

El panel de administración solo es visible para usuarios con rol administradores. Se divide en cuatro pestañas con las que se pueden aplicar la mayor parte de las modificaciones que requiera la página sin necesidad de acceder a base de datos. (Ver figura 3.11).

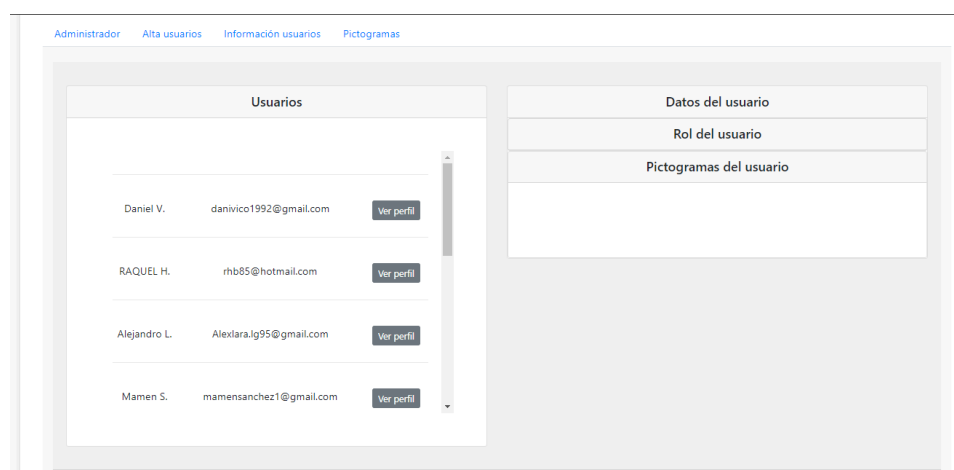


Figura 3.11: Vista de la página Administración de la versión final

La pestaña administrador es idéntica a la de perfil de usuario, explicada anteriormente. Para el alta de usuarios se reutiliza la del registro, añadiendo el campo rol. La pestaña de información de usuarios es la que se puede ver en la figura 3.11. En la parte izquierda aparecen los usuarios de la página y cuando se quiere ver el perfil de uno de ellos, aparece su información en la parte derecha incluyendo los pictogramas que tenga subidos. Por último, pictogramas se corresponde con la información de las imágenes que usa la página. En ella se pueden añadir nuevos pictogramas y categorizarlos antes de añadirlos, y borrar existentes.

## 3.6. Conclusiones

En este capítulo hemos explicado el diseño de la interfaz gráfica siguiendo el proceso iterativo citado en el capítulo del Diseño Centrado en el Usuario.

Aunque fueron varios los bocetos e ideas iniciales, se ha optado por seguir el diseño más cómodo y sencillo pensando siempre en el punto de vista de los usuarios finales. No solo se ha pensado en las personas que la van a utilizar si no también en aquellas que podrían quedarse a su cargo en el futuro, para mejorarla y mantenerla, para que pueda mejorar su contenido continuamente sin necesidad de tener conocimiento sobre el código en el que se basa.

Ahora que ya tenemos la interfaz diseñada, podemos pasar a explicar que tecnologías hemos utilizado para el desarrollo de la página, y cuales de ellas se han mantenido y usado, y cuáles de ellas han sido descartadas.



# Capítulo 4

## Tecnologías

*“Sabemos lo que somos, pero no lo que podemos llegar a ser.”*  
— William Shakespeare

En este capítulo vamos a hablar sobre las tecnologías utilizadas durante el desarrollo de nuestro proyecto, tanto en la redacción de la memoria como en la implementación de la aplicación.

### 4.1. HTML

HTML (HyperText Markup Language) es un lenguaje de marcas que se dedica a la elaboración de páginas web, darles estructura y contenido.

Este lenguaje nació en 1991 bajo el estándar del World Wide Web. Fue desarrollado por Tim Berners-Lee y algunos de sus compañeros del CERN en Suiza. Desde ese momento se ha seguido desarrollando el estándar hasta llegar al actual HTML5 (Luján-Mora, 2001), (Gutiérrez, 2019).

Las páginas se pueden clasificar en dos tipos dependiendo de cómo se generan en el servidor y de cómo se visualizan en el cliente. Éstas pueden ser estáticas o dinámicas.

Las páginas estáticas poseen un contenido fijo independientemente de quien las consulte, es decir, todos los usuarios reciben la misma información. Son aquellas que no poseen código de scripts, applets o plug-ins. Por otro lado, las páginas dinámicas poseen un contenido variable, es decir, dependiendo del usuario que la utilice, el contenido que se muestre en ella será distinto. Éstas últimas ejecutan código en el equipo del usuario.

Las características anteriores se pueden combinar.

Gracias a todas las etiquetas que nos ofrece este lenguaje, las páginas podrán ser personalizadas con distintas fuentes y colores que den forma a los textos. Además, permite incluir imágenes, listas o tablas. También le damos

el formato al contenido y decidimos la distribución de los elementos a lo largo de la página.

En HTML la mayoría de etiquetas aparecen por parejas, delimitando la parte del documento en la que afecta su acción. También pueden aparecer de forma individual como es el caso de `<img>`, el cual utilizamos para insertar una imagen. Todas estas marcas comienzan con el símbolo '`<`' y terminan con '`>`', y entre estos símbolos se pone el nombre de la etiqueta. Aquellas que se usan para delimitar, poseen el mismo nombre que las de inicio pero llevan delante el símbolo '`/`'.

Este lenguaje no distingue entre mayúsculas y minúsculas, pero se considera una buena práctica poner las etiquetas en mayúsculas para facilitar su legibilidad.

La estructura básica de este lenguaje son los elementos, que se caracterizan por tener dos propiedades: **atributo** y **contenido**. Cada elemento tiene una etiqueta de inicio y otra de fin y entre ellas están situados los atributos. Éstos son, por norma general, pares nombre-valor, separados por `=` y rodeados por comillas dobles o simples.

## 4.2. CSS

CSS (Cascading Style Sheets) es un lenguaje de diseño gráfico que sirve para crear la presentación de un documento estructurado escrito con lenguaje de marcas.

Se utiliza habitualmente en sitios codificados con HTML o JavaScript para hacer que éstos se vean más atractivos. También se utiliza para crear interaces de usuario en aplicaciones web y en aplicaciones móviles.

Al separar la estructura del documento del diseño se crea menos código y mas transparente. Esto hace que el código se estructure en bloques lógicos claros que permitan una mejor lectura y mejora los tiempos de carga de las páginas.

También facilita las actualizaciones al tener todas las instrucciones de diseño y formato en una hoja de estilo separada del documento HTML, por lo que todas las modificaciones se pueden realizar de forma centralizada.

Los documentos CSS normalmente constan de una serie de instrucciones donde se definen como se representan los elementos de la página. Para identificar las etiquetas HTML se utiliza `'.'` más la clase o `'#'` cuando usamos los identificadores de la etiqueta.

CSS no distingue entre mayúsculas y minúsculas.

Para juntar CSS con HTML, las instrucciones de estilo se pueden añadir directamente en las etiquetas HTML o en la cabecera del documento. Las instrucciones se escriben en un archivo externo y se referencian desde cada archivo HTML (Schulz, 2008).

## 4.3. XAMPP

XAMPP es un paquete de software libre que se utiliza para la gestión de aplicaciones o bases de datos MySQL. Consiste en un servidor HTTP de Apache, una base de datos MariaDB e intérpretes para los lenguajes PHP y Perl.

Este programa nos permite reducir el tiempo que lleva instalar todos los componentes que necesitamos para realizar las pruebas de la aplicación que estamos desarrollando, agrupando todos los componentes en un mismo sitio. Este paquete ofrece la posibilidad de crear tablas y bases de datos de forma fácil y es capaz de interpretar páginas dinámicas.

XAMPP está disponible tanto para Windows como para Mac OS X y GNU/Linux (Wikipedia, 2018c).

### 4.3.1. Apache

Apache es el principal componente que utilizamos de XAMPP. Es un servidor web de código abierto cuyo uso principal es enviar páginas web estáticas y dinámicas en la World Wide Web. Localmente, lo hemos utilizado porque permite previsualizar el contenido y la funcionalidad de nuestra aplicación web, y así ir probando cada modificación que hemos ido realizando a lo largo del diseño. También será utilizado en la versión final ya que permite establecer el contenido de forma clara y fiable. (Apache, 2019)

### 4.3.2. MySQL

MySQL es el sistema de gestión de base de datos que utiliza XAMPP y por tanto, es el que hemos utilizado también en nuestro proyecto. La lectura de los datos se hace de forma ágil, y además, como en las aplicaciones web la concurrencia que se da en la alteración de los datos es muy reducida, hace que esta base de datos sea ideal para este tipo de desarrollo.

MySQL es sistema de software libre mas popular para gestionar bases de datos SQL. Ha sido desarrollado, distribuido y gestionado por Oracle Corporation. (Oracle, 2019)

Las características mas importantes son:

- Los datos se guardan de forma estructurada.
- Los datos se guardan en tablas separadas.
- Es Open Source, por lo que cualquiera puede descargar el software y utilizarlo sin pagar por ello.
- Es un sistema muy rápido, de confianza, escalable y fácil de usar.
- Es un sistema con un esquema cliente/servidor.

### 4.3.3. PHP

PHP (Hypertext Preprocessor) se utiliza para codificar la parte que se encuentra en el servidor.

Es un lenguaje que permite ser desplegado en la mayoría de servidores y sistemas operativos, por ello es muy utilizado para la creación de aplicaciones web.

Permite realizar conexiones sencillas tanto a servidores de base de datos (MySQL en nuestro caso) o servicios externos, como es el caso de los servicios REST que utilizamos en nuestro proyecto.

Su funcionamiento se basa en una petición que hace el cliente al servidor para mostrar una página web, haciendo que el servidor ejecute un intérprete de PHP. La respuesta al procesar el script correspondiente, es enviada al cliente a través del intérprete (Mehdi Achour, 2019).

## 4.4. GitHub

GitHub<sup>1</sup> es una plataforma con un servicio cloud de repositorios que sirve para que los desarrolladores puedan administrar su código, además de llevar un registro de los cambios y un control de versiones. Fue creada en 2007 por Chris Wanstrath, P. J. Hyett, Tom Preston-Werner y Scott Chacon. Actualmente es propiedad de Microsoft, que la adquirió en octubre de 2018.

Las principales funcionalidades que ofrece GitHub son el control de versiones, mediante el cual se pueden registrar los cambios que se producen en el software además de quién ha realizado dichos cambios, pudiendo volver a una versión anterior si fuera necesario. Gracias a esta funcionalidad se puede continuar desarrollando un proyecto por otra rama desde una versión.

Los repositorios son de tipo Git y pueden ser de un único desarrollador o de varios. Su funcionamiento consiste en tener una copia del código en local, que es donde se modifica el código y cuando se publican los cambios se cambia la versión almacenada en la nube a la versión que acabamos de publicar, haciéndola visible al resto de desarrolladores. De esta forma se puede continuar desarrollando un proyecto y volver a una versión anterior si nos hemos equivocado o no funciona (Miró, 2017).

## 4.5. JavaScript

JavaScript es un lenguaje de programación cuyo principal uso es crear páginas web dinámicas. Se entiende por página web dinámica aquella que incorpora efectos o animaciones que se activan cuando se pulsan botones por

---

<sup>1</sup><https://github.com/>

ejemplo.

Es un lenguaje interpretado, por lo que no hace falta compilar el programa para ejecutarlo. Por lo tanto se puede probar en cualquier navegador sin necesidad de realizar procesos intermedios.

Este lenguaje nació a principio de los 90 por la necesidad de que el lenguaje de programación se ejecutara directamente en el navegador del usuario debido a la baja velocidad de la época (28.8 kbps), lo que generaba un problema al tener que hacer llamadas al servidor que tardaban mucho tiempo.

JavaScript se puede incluir en documentos XHTML de tres formas distintas.

La primer forma es incluir el código JavaScript entre etiquetas `script`. Se puede incluir en cualquier parte del documento, pero se recomienda incluir el código en la cabecera, dentro de la etiqueta `head`.

También se puede definir en un archivo externo del tipo JavaScript, con extensión `.js` que se enlaza desde los documentos XHTML.

El último método consiste en introducir JavaScript dentro del código HTML de la página.

Este método es el que menos se utiliza ya que complica el mantenimiento del código (Eguíluz Pérez, 2012).

## 4.6. jQuery

jQuery es una biblioteca (API) de JavaScript que ayuda a hacer más sencilla la tarea de crear y desarrollar una página o aplicación web. Esta herramienta permite la creación de los sitios web en un menor tiempo que si se emplease únicamente JavaScript, además de hacer que el código sea menor y más eficiente. La forma de hacer esto es que nos posibilita el acceso a los elementos del DOM del sitio web, es decir, a las imágenes, textos y enlaces entre otros, para poder modificar su diseño, crear eventos sobre dichos elementos o hacer uso de Ajax que explicaremos posteriormente. (Dominguez, 2018)

### ■ Características más importantes

La característica que más notoriedad ha tenido es su capacidad de adaptación a cualquier plataforma, es decir, se adecua a cualquiera de los navegadores existentes (Chrome, Firefox, Safari, etc).

Gracias a que favorece el manejo de Ajax, permite una constante comunicación entre el servidor y la página, sin necesidad de una constante carga de la web. El inconveniente de esto es que cada navegador ejecuta distintamente la API de Ajax.

Como hemos dicho anteriormente, posibilita operar sobre los elemento del DOM, además de crear animacione, alertas o eventos sobre dichos elementos.

(B., 2019)

## 4.7. Ajax

Ajax<sup>2</sup> es una tecnología que es una combinación de cuatro ya existentes, que son HTML, DOM (Document Object Model), XMLHttpRequest y XML. Se desarrolló con el fin de agilizar la comunicación entre la página y el servidor durante los desarrollos web. Esto permite la recuperación de datos del servidor y su uso por parte de JavaScript. Esto hace que no haya una constante necesidad de recargar la página web por completo, lo cual hace que el usuario no note las demoras que podría ocasionar la carga de los datos.

## 4.8. REST

REST (Representational State Transfer) es un estilo de arquitectura software que se utiliza para crear servicios web, que sirven para que las aplicaciones puedan intercambiar datos entre ellas. Estos servicios proporcionan interoperatividad entre diferentes sistemas en Internet.

Para considerar servicios REST, los servicios web deben cumplir ciertas premisas que vamos a explicar a continuación.

- **Cliente/Servidor.** Como los servicios web son de tipo cliente servidor, se define una interfaz de comunicación que separa las responsabilidades de ambas partes.
- **Sin estado.** Estos servicios no mantienen el estado asociado al cliente, por lo que cada petición es independiente de las demás y todas las llamadas a un mismo servicio serán idénticas.
- **Cache.** El contenido de los servicios REST se cachea permitiendo que tras la primera petición al servicio las siguientes pueden utilizar la cache si fuese conveniente.
- **Servicios Uniformes.** Todos los servicios comparten la misma forma de invocarse y unos métodos uniformes. Los métodos que se utilizan son GET, POST, PUT y DELETE.
- **Arquitectura de Capas.** Los servicios REST están pensados para ser escalables y que un cliente al realizar una petición no es capaz de distinguir si la respuesta se la está proporcionando una cache intermedia, o qué servidor, ya que por ejemplo existe un sistema de balanceo de carga que redirige el tráfico entre los servidores (Caules, 2013).

---

<sup>2</sup><https://es.wikipedia.org/wiki/AJAX>

## 4.9. SQL

SQL (Structured Query Language) es un sistema utilizado para administrar sistemas de base de datos. Fue diseñado para la gestión de bases de datos relacionales.

Es un lenguaje declarativo de alto nivel cuyas características vamos a enumerar a continuación.

**Lenguaje de definición de datos.** Proporciona comandos para crear esquemas de relación, modificarlos y borrarlos.

**Lenguaje interactivo de manipulación de datos.** Incluye un lenguaje de consultas basado en el álgebra relacional.

**Integridad.** Posee comandos para poder especificar las restricciones de integridad que deben cumplir los datos almacenados.

**Control de transacciones.** Tiene mecanismos para definir el comienzo y finalización de las transacciones (Wikipedia, 2019).

## 4.10. Bootstrap

Bootstrap es un framework que se creó primeramente como una solución para twitter, pero más tarde se publicó como Open Source para que el público pudiera hacer uso de él.

Este framework engloba las tres tecnologías que más se usan para el diseño de páginas web, es decir, HTML, CSS y JavaScript. Su principal utilidad es la de favorecer la creación y el diseño de páginas web, de forma que ofrece un gran control sobre la visión de las interfaces web en los distintos dispositivos (pcs, tablets y móviles).

Su uso es sencillo ya que además de poder descargarlo para agregarlo a nuestro sitio web, también permite su uso mediante links para las hojas de estilos (CSSs) o para los scripts que se crean con JavaScript (Benites, 2018).

## 4.11. JSON

JSON es un formato para intercambiar datos, que surgió a principios de la década del 2000 con la finalidad de definir un formato que fuera más ligero y rápido que los que había en aquel momento. JSON facilita que las máquinas puedan interpretarlo. Su base es JavaScript, pero aunque es un formato independiente, hace uso de estándares conocidos por otros lenguajes, por ello es muy utilizado para el intercambio de datos.

Este formato permite el intercambio de datos entre distintas máquinas que utilizan lenguajes de programación diferentes, ya que actualmente la gran mayoría de los lenguajes dan soporte al intercambio de datos en este formato. Otra de las ventajas es que el volumen de los datos necesarios para

intercambiar información es mucho menor que en otros lenguajes como por ejemplo XML.

Está formado por dos estructuras, la primera es una colección de pares nombre/valor o lo que es lo mismo, una colección de objetos. La segunda, es una lista ordenada de valores formando así lo que se conoce en otros lenguajes como array o vector. Que siga estas dos estructuras se debe a que todos los lenguajes (C, Java, Python, etc) también las tienen implementadas de una u otra manera (jso, 2015).

## 4.12. Servicios Web

Los servicios web sirven de intercomunicadores entre distintas máquinas conectadas a una red. La forma de interactuar con los servicios se basa en peticiones y respuestas entre dos máquinas en las que una de ellas ejerce como cliente (será la que haga las peticiones) y la otra será el servidor (será quien facilite la respuesta con los datos) (web, 2015).

**SOAP:** Siglas de Simple Object Access Protocol. Fue el primer tipo de servicio web que apareció. SOAP es un protocolo que establece la forma en que deben de realizarse las comunicaciones. La base de este tipo de servicio es el lenguaje XML.

**REST:** Son las siglas en inglés de Representational State Transfer. La base de este tipo de servicios es el uso del protocolo de comunicación HTTP, ya que todos los sistemas lo soportan. Su característica principal es que carece de estados, por lo que no puede recordar las solicitudes que un cliente haya hecho anteriormente ya fuesen satisfactorias o no. Cuando la máquina cliente quiera comunicarse con él, tendrá que enviarle un token con toda la información necesaria. Un token no es otra cosa que una cadena de texto, que puede ser interpretada por el servidor. Las respuestas de este tipo de servicios vienen dadas en formato JSON, del que ya hemos hablado anteriormente.

## 4.13. Java

Java<sup>3</sup> es un lenguaje de programación orientado a objetos publicado en 1995 por Sun Microsystems (posteriormente comprada por Oracle). Su intención era que el código sólo fuera necesario escribirlo una vez y se pudiera ejecutar en cualquier dispositivo.

La sintaxis es similar a la de C o C++, ya que se deriva principalmente de estos lenguajes, pero al contrario que en ellos en Java se disponen de menos utilidades de bajo nivel al ser un lenguaje de mas alto nivel que los citados anteriormente. Una funcionalidad destacada que aporta java es el recolector de basura. Consiste en un proceso que se encarga de liberar la memoria que

---

<sup>3</sup><https://www.java.com/es/>



ocupan los objetos que ya no se encuentran referenciados. De este modo es el programador el que se encarga de crear los objetos pero el sistema es el encargado de liberar la memoria evitando fugas.

Este lenguaje se ejecuta en la JVM (Java Virtual Machine), que es una máquina virtual escrita en Java que se encarga de ejecutar instrucciones en formato Byte Code, producido al compilar archivos java. Cabe destacar que no es una máquina virtual como a las que estamos acostumbrados, que ejecutan un sistema operativo completo, sino que la máquina virtual recibe el código Byte Code procedente de la compilación de JAVA y lo compila al código necesario para el sistema en el que se está ejecutando. De esta forma por ejemplo se puede utilizar un mismo código java para diferentes sistemas operativos.

Java se utiliza tanto en dispositivos móviles, navegadores web, aplicaciones de escritorio o aplicaciones en las que se necesite recurrir a la programación concurrente.

También existen servidores como Tomcat<sup>4</sup>, que es un servidor web desarrollado con el lenguaje JAVA, lo que posibilita su uso en diferentes sistemas operativos. Fue diseñado como para ejecutar servicios web hechos en java y utiliza el contenedor de servlets Catalina.

## 4.14. Jersey

Es una biblioteca de software libre que implementa un cliente rest que sirve para crear servicios rest con Java. Esta biblioteca permite llamar a métodos de java como si fueran servicios web<sup>5</sup>.

Para proveer de funcionalidad REST a una clase se emplean anotaciones en ésta. Las principales anotaciones son las siguientes:

- `@Path` : Indica la URL de la clase o el método a la que se van a realizar las peticiones.
- `@Get` : Indica que el método anotado procesará las solicitudes HTTP GET.
- `@Post` : Indica que el método anotado procesará las solicitudes HTTP POST.
- `@Produces` : Indica el tipo de formato que producen los métodos como respuesta al cliente. Los tipos pueden ser por ejemplo texto plano, JSON o HTML.
- `@QueryParam` : Sirve para enlazar un parámetro de una petición HTTP con un parámetro del método Java que tiene la anotación.

---

<sup>4</sup><https://tomcat.apache.org/>

<sup>5</sup><https://jersey.github.io/>

### 4.15. GNU/Linux

Es un sistema operativo, es decir, un conjunto de programas que permiten que el usuario y el ordenador puedan interactuar, además de permitir al usuario ejecutar programas. En nuestro caso este será el sistema operativo que se encuentre en nuestro servidor, donde se hospedará la página web.

Un sistema operativo está compuesto por distintos programas necesarios para el usuario pueda enviar instrucciones y realizar acciones sobre el ordenador. En este caso Linux<sup>6</sup> es el núcleo del sistema operativo, lo que podríamos comparar con el corazón ya que sin el kernel no se podría ejecutar ninguna tarea, ya que su principal función es que el software y el hardware interactúen entre ellos. Algunas de sus funciones son gestionar la memoria de los programas, administrar el tiempo que cada proceso utiliza el procesador o gestionar los accesos a los periféricos. Este núcleo lo empezó a desarrollar Linus Torvalds en la década de los 90 como un proyecto de software libre y actualmente es usado ampliamente por ejemplo en dispositivos móviles. (Lin, 2019)

El proyecto GNU<sup>7</sup> empezó a desarrollarlo la Free Software Foundation fundada por Richard Stallman a mediados de la década de los 80 con el objetivo de crear un sistema operativo de software libre. Como no tenían desarrollado un núcleo decidieron añadir el kernel de Linux, fundando así GNU/Linux.

### 4.16. Conclusiones

En este capítulo hemos estudiado las tecnologías que se utilizan actualmente para crear bases de datos, realizar programación web, programación orientada a objetos o creación de servicios REST.

Estos conocimientos nos permiten implementar el diseño creado anteriormente, por lo que el próximo paso que vamos a seguir va a ser desarrollar la aplicación utilizando el diseño y las tecnologías explicadas anteriormente.

---

<sup>6</sup><https://www.linux.org/>

<sup>7</sup><https://www.gnu.org/home.es.html>

# Capítulo 5

## PictoPad

*“La educación es el arma más poderosa que puedes usar para cambiar el mundo.”*

— Nelson Mandela

En este capítulo se va a hablar del componente principal del proyecto, la aplicación web PictoPad (<https://holstein.fdi.ucm.es/tfg-pictoagenda/>), la cual integra el servicio de predicción que se explicará posteriormente.

Hablaremos sobre toda la arquitectura interna de la pagina web que hemos desarrollado, ofreciendo una visión en profundidad del sistema, los componentes y funcionalidades implementados. Es decir, toda aquello que formará el diseño tanto de forma lógica (funcionamiento) como física (diseño).

Nuestro diseño se encuentra estructurado siguiendo el modelo vista-controlador.

### 5.1. Modelo-Vista-Controlador (MVC)

El MVC es un patrón de diseño software que permite separar la forma en la que se presentan los datos al usuario de la forma en la que se manejan y de cómo se accede a los datos.

Por un lado tenemos la parte lógica de la aplicación, y por otro la lógica de la parte visual. Gracias a ello podemos tener separados todos los componentes que forman nuestra aplicación, y esto implica, que a la hora de hacer modificaciones en cualquier parte del código fuente, no sea necesario tener que modificar el resto (ver Figura 5.1). (Hernandez, 2019).

**Modelo:** El modelo contiene los datos de la aplicación, pero no tiene información sobre como mostrar esos datos al usuario. Puede recibir peticiones sobre su estado actual o para cambiarlo.

**Vista:** se encarga de la visualización de los datos. Representa toda la interfaz con la que el usuario interactúa y es el componente encargado

de gestionar la salida gráfica o textual de toda la aplicación. La vista obtiene los datos del modelo para generar la interfaz.

**Controlador:** es la parte que sirve de interconexión entre el modelo y la vista. Se encarga de obtener el evento de la vista y notificar al modelo la acción del usuario para que el modelo pueda cambiar su estado en función de la acción.

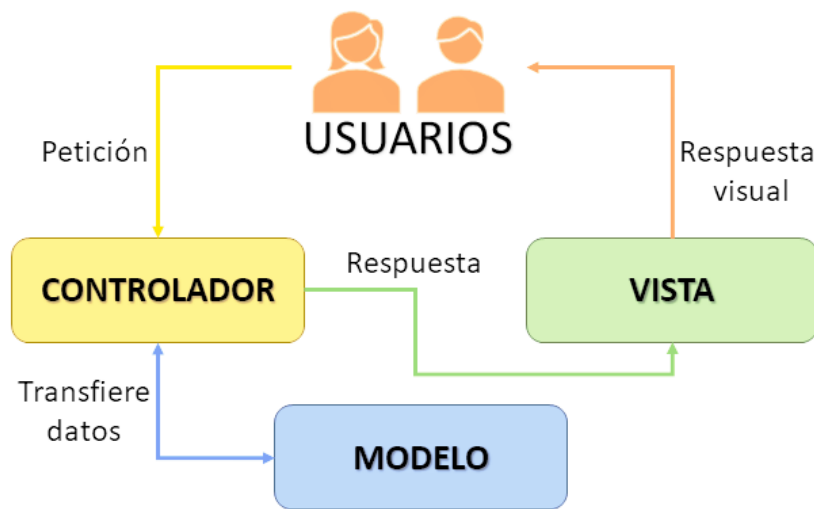


Figura 5.1: Modelo-Vista-Controlador

## 5.2. Bases de Datos

La base de datos es el banco de todos los datos que necesitamos para la página. Nosotros hemos optado por utilizar MySQL para implementarla, ya que las lecturas que realiza son rápidas. Es una base de datos relacional, es decir, hace uso tanto de la teoría de conjuntos como de la lógica predictiva o de primer orden (ver Figura 5.2).

**Pictogramasusuario:** esta tabla está creada para almacenar los pictogramas subidos por un usuario. Tiene tres atributos, el primero es su clave identificativa, el segundo es el nombre del pictograma (incluye el formato de la imagen (.jpg o .png)) y por último el id del usuario, que se utiliza para relacionarlo con la tabla usuarios.

**Pictogramas:** se utiliza para almacenar los pictogramas obtenidos de la web de ARASAAC. Al igual que la tabla anterior, tiene tres atributos,

su clave identificativa, el nombre del pictograma, que al igual que en la tabla `pictogramasusuario`, contiene el formato de la imagen. En último lugar está la categoría a la que se ha asociado la imagen.

**Usuarios:** es la más importante, ya que almacena los datos de todos los usuarios registrados. Guarda los datos básicos de un usuario, su rol y su contraseña encriptada. Los roles serán explicados más adelante.

**Frasesusuarios:** esta tabla almacena las frases que los usuarios quieran guardar para verlas en cualquier otro momento. Sus atributos son, el identificador de la frase, el identificador de usuario el cual permite la relación de esta tabla con la de usuarios, el tipo de traducción (tp será de texto a pictogramas y pt será el caso contrario). La frase y las imágenes correspondientes a la traducción y por último la fecha de cuando se ha realizado la traducción.

**Predictor:** almacenará todos los pares de palabras consecutivos que aparezcan en las frases que creen los usuarios. Sus atributos, al igual que en tablas anteriores serán el identificador de cada predicción, las palabras actual y siguiente y, al final, la frecuencia con la que se usa dicho par.

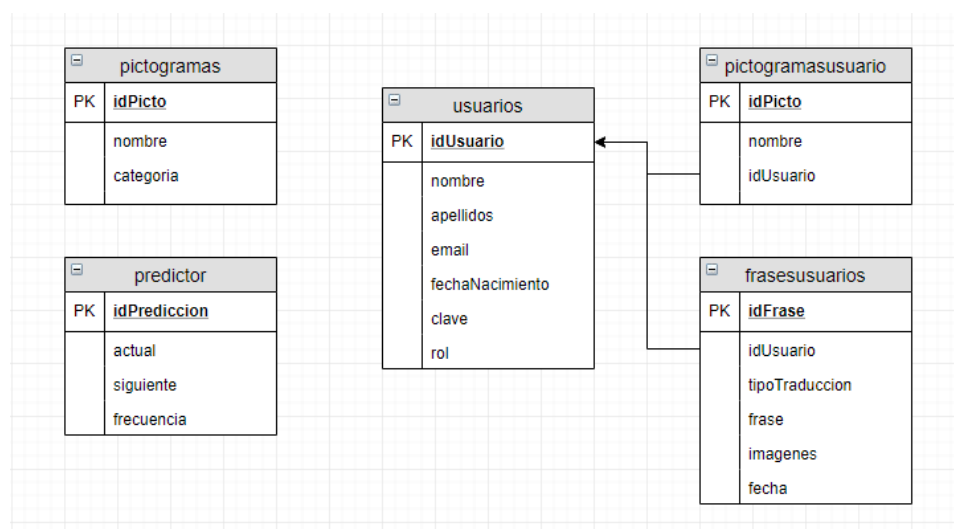


Figura 5.2: Diagrama representativo de la Base de Datos

### 5.3. PictoPad

A fin de que la aplicación pueda ser utilizada en distintos dispositivos, se ha realizado un diseño responsive. Éste inicialmente se intentó lograr uti-

lizando las hojas de estilos CSS con media queries, pero finalmente se optó por utilizar Bootstrap para conseguir un diseño más equitativo y apto para todos los dispositivos.

La página principal se encuentra dividida en cuatro áreas: traductor, editor, buscador y sugerencias. A parte, también existe un perfil de usuario, y una consola de administración. Lo que se buscaba era diseñar una interfaz sencilla puesto que la aplicación está orientada a personas que padecen el síndrome del espectro autista, sobre todo, está pensada para que puedan utilizarla niños y niñas. Para ello, se hace uso de los pictogramas que se encuentran en ARASAAC.

El que se hayan creado cuatro áreas de trabajo en la página principal, se debe a que se buscaba que la información estuviese ordenada y separada, para que visualmente tenga una disposición más sencilla, y que en un solo vistazo se pueda tener acceso a todas las partes. A continuación, se describe cada una de las partes que forma esta aplicación.

## 5.4. Sección Traductor

El traductor es una de las dos principales áreas con las que el usuario puede trabajar. En esta zona el usuario puede introducir cualquier texto en castellano y se traducirá en pictogramas.

Para esta sección se hace uso del servicio de predicción implementado en PICTAR<sup>1</sup>. Este servicio nos devuelve el resultado de traducir la frase del usuario en formato JSON. Para obtener el contenido, utilizamos la función *file\_get\_contents* y con *json\_decode* lo decodificamos, es decir obtenemos un array (lista de string) que podremos recorrer para obtener los números que identifican los pictogramas.

Una vez se tienen todos los identificadores, se utiliza el servicio de recuperación de pictogramas<sup>2</sup> para obtener el pictograma asociado a cada identificador. Como cada palabra puede estar representada por más de un pictograma, para cada imagen se muestra un carousel de imágenes con dos flechas para que pueda ir cambiándolas. Para ello nos movemos por las posiciones del array de imágenes perteneciente a cada palabra. Cuando una palabra no contiene un pictograma asociado, lo que se hace es mostrar la palabra, para que no quede en ningún momento la traducción a medias. Por otro lado, todas las imágenes tienen la opción de ser eliminadas por si en algún momento el usuario quiere reducir la frase o sobre todo para quitar los determinantes, y de esa forma hacer más sencillas las oraciones.

En esta sección aparecen cuatro botones. Traducir es el que permite al usuario traducir la frase. Para que el usuario vea que se ha ejecutado su petición de traducción, se muestra un mensaje informando de que la traduc-

<sup>1</sup><http://hypatia.fdi.ucm.es:5223/PICTAR/traducir/frase>

<sup>2</sup><http://hypatia.fdi.ucm.es/conversor/Pictos/identificador>

ción está en proceso. Para este mensaje se usa una alerta y con las funciones show y hide de jQuery se controla cuando se tiene que mostrar u ocultar la alerta. El botón de guardar solo aparece cuando el usuario está logueado en la aplicación y le permite guardar la frase que ha creado para verla en cualquier otro momento en su perfil. El botón descargar permite exportar la traducción en formato PDF, para que así se pueda imprimir o guardar. Para poder exportarlo en PDF se ha usado la librería FPDF. El último botón, limpiar, permite al usuario borrar tanto la frase que ha escrito como la traducción para que así pueda realizar una nueva de forma más clara.

Entre la parte donde se introduce el texto y los botones aparece un hueco donde aparecerá la predicción del texto. La implementación de los resultados de predicción se basa en botones con las palabras que tienden a aparecer detrás de la última palabra que se haya escrito. El hecho de que sean botones se debe a lo mismo que las imágenes, y es que al estar pensado para que se pueda utilizar en dispositivos como tablets, resulta más cómodo, sencillo y usable que sean botones (ver Figura 5.3).

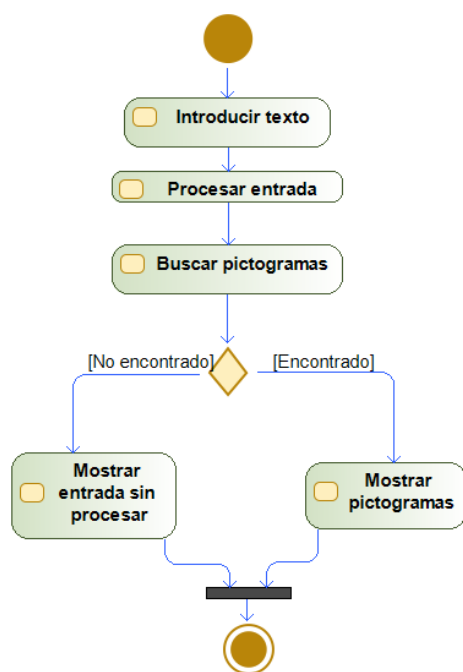


Figura 5.3: Diagrama de actividad de la sección Traductor

## 5.5. Sección Editor

Esta sección aparece inicialmente vacía, simplemente aparece el espacio para los pictogramas y para la traducción. El usuario tendrá que buscar

los pictogramas con los que quiere crear la frase en el buscador o en las categorías. Una vez encuentre el pictograma deseado y haga click sobre él, este aparecerá en el editor. Se ha elegido esta forma de implementarlo ya que para usuarios con TEA y niños es más sencillo esto y no tener que arrastrarlas por la pantalla.

Según se vayan agregando imágenes, al mismo tiempo se irá creando de forma automática la traducción de la frase que se está creando. El texto con la traducción aparecerá únicamente en la parte posterior del editor en modo lectura. Todas las imágenes tienen la opción de poder ser eliminadas lo cuál hace que el editor sea bastante versátil y editable.

Al igual que en la sección anterior, aparecen los botones de guardado, descarga y limpia, con las mismas funcionalidades descritas anteriormente.

## 5.6. Sección Buscador

Esta sección aparece en la parte superior derecha. Está implementada con una búsqueda en la base de datos. Para ello se han descargado los 15394 pictogramas a color y en castellano disponibles en la página de ARASAAC y se han guardado en una tabla de base de datos, junto con la categoría que le ha sido asignada.

La búsqueda es constante gracias al uso de javascript, jquery y Ajax, lo cual permite realizar consultas continuas sin necesidad de recargar la página. Ésta búsqueda está implementada de forma que busque los pictogramas que comiencen por las letras que el usuario vaya introduciendo. Esto se realiza con el evento onchange que tiene el input de la búsqueda.

Los pictogramas están implementados como botones, para que, como hemos explicado en la parte del editor, una vez que se pinche en la imagen, automáticamente aparezca en la sección del editor.

Inicialmente se había limitado la búsqueda por la posibilidad de que apareciesen demasiadas imágenes, pero finalmente se ha optado por quitar ese límite y hacer que el grid donde se muestran tenga una barra de scroll, es decir, una barra con la que desplazarnos a través de todos los resultados, consiguiendo que el efecto de la búsqueda no se haga pesado, ni muy cambiante, además de reducir así el espacio que ocupa la sección.

Como el número de pictogramas es limitado, se pueden dar casos en los que no se encuentren imágenes asociadas a una palabra. Para ello, se ha implementado una alerta con jQuery para que avise al usuario cuando no haya resultados coincidentes con la búsqueda (ver Figura 5.4).



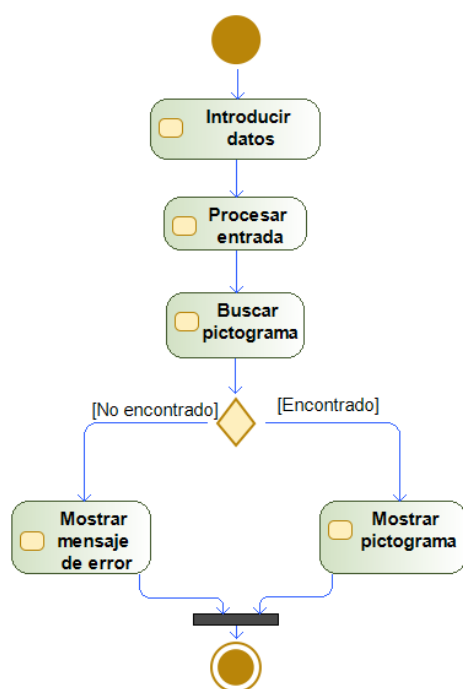


Figura 5.4: Diagrama de actividad de la sección Buscador

## 5.7. Sección Sugerencias

La sección sugerencias está dividida en dos partes, la primera se encuentra justo debajo del título de sugerencias donde se puede ver que aparece un espacio. Ese hueco es el que se encuentra reservado para el predictor de pictogramas. Se ha mantenido en la parte derecha para que no resulte visualmente cambiante para los usuarios, ya que esa parte es donde se podrán encontrar todas las imágenes. En el área posterior hay un dropdown con todas las categorías. Una vez se seleccione una categoría automáticamente se cargaran los pictogramas que se encuentren asociados a ella, y una vez se elija uno de ellos, aparecerá en el grid de la sección de edición.

Cuando el usuario esté logueado en la página, a parte de las categorías por defecto, aparecerá una nueva llamada *Mis pictogramas* en la que el usuario podrá encontrar las imágenes que ha subido.

## 5.8. Perfil de usuario

Esta sección solo se encuentra disponible cuando el usuario a iniciado sesión en la página, o cuando se acaba de registrar ya que automáticamente se queda ya con esa sesión iniciada.

Esta página esta dividida en dos partes. En la izquierda, el usuario puede ver sus datos personales pero no modificarlos. Lo que sí puede cambiar es su contraseña por una nueva, y para ello utilizamos la doble comprobación de la nueva contraseña. Para que las contraseñas queden almacenadas encriptadas utilizamos la función `password_hash` con `PASSWORD_DEFAULT` como método de encriptación ya que utiliza la forma más segura. Y para la desencriptación utilizamos la función `password_verify`.

En el lado derecho de la página se encuentra el apartado de pictogramas del usuario. En su parte superior puede ver las imágenes que ha subido anteriormente y que puede utilizar en el editor, incluyendo la opción de eliminarlos. En la parte inferior puede añadir nuevos pictogramas.

En la parte inferior el usuario puede encontrar las frases que ha escrito y guardado, filtrandolas por la fecha en la que las escribio, lo que le permite disponer de una agenda con recordatorios.

## 5.9. Panel de administración

Esta página se ha creado para facilitar el trabajo a los administradores de la página, y con el fin de que se puedan hacer las funciones más comunes desde una interfaz sin necesidad de acceder al servidor o a la base de datos. El panel de administración tiene un menú formado por cuatro opciones:

La pestaña *Administrador* representa el perfil del usuario administrador. Es idéntica al perfil de usuario normal.

Alta usuarios es como la página del registro, pero aquí el administrador puede elegir con que rol quiere añadir a otro usuario.

En información de usuarios, aparecerá la lista de usuarios registrados en la página. Pulsando el botón ver perfil, aparecerá en la derecha toda la información del usuario elegido. Además se le podrá cambiar el rol y también puede ser eliminado. Aquí el administrador puede ver los pictogramas que tiene el usuario, además de poderlos borrar si lo cree preciso.

El apartado pictogramas está pensado para poder añadir nuevos pictogramas de uso común. Esto será útil para que se puedan ir añadiendo de forma sencilla y rápida los nuevos pictogramas que se creen en ARASAAC. Además antes de subirlo se puede categorizar para que aparezca en la página principal, aunque no es obligatorio. No solo se pueden añadir pictogramas si no que también podrán ser eliminados para cuando queden obsoletos o sean modificados por nuevos.

## 5.10. Predictor

Como se ha explicado en el capítulo de tecnologías, hemos hecho uso del texto predictivo para generar palabras o pictogramas en base a lo que el

usuario haya introducido anteriormente con el fin de simplificarle la creación de las oraciones. Para ello cada miembro ha diseñado un predictor distinto que se explican a continuación.

### 5.10.1. Predictor basado en análisis de frecuencias

Este predictor esta diseñado en la propia página web y por lo tanto está creado haciendo uso en gran medida del lenguaje php (ya explicado también en el capítulo de tecnologías). El diseño de éste, consiste en una serie de consultas a una tabla de base de datos llamada **predictor**. Esta tabla, como se ha definido anteriormente en este capítulo tiene cuatro atributos, pero para explicar el funcionamiento solo necesitamos hablar de tres de ellos. Dos de estos forman un par, y es que el atributo siguiente depende por completo del actual. A continuación se explica como funciona esta forma de predicción:

#### 5.10.1.1. Funcionamiento

La idea base es tener una tabla en la que guardar los pares de palabras que se utilizan de forma consecutiva, y además, tener un atributo más que es el encargado de llevar la cuenta de la frecuencia con la que se usa cada par. Por lo tanto, cada vez que el usuario cree una frase en el traductor o en el editor, se comprobará si cada par que aparece en dicha oración se encuentra ya guardado en la base de datos, si es así, se incrementa la frecuencia del par en uno, si por el contrario no existe, se guardará el nuevo conjunto de palabras en la base de datos con frecuencia inicial en uno. La misma base de datos sirve tanto para guardar los pares de palabras que se usan en los textos como las parejas que contienen los nombres de imágenes que aparezcan de forma consecutiva. Se adjunta diagrama con la representación de las etapas del predictos (ver Figura 5.5).

### 5.10.2. Predictor basado en Cadenas de Markov

Para realizar la tarea de traducción estuvimos estudiando diferentes opciones para obtener la predicción de los pictogramas. Desde un primer momento estuvimos estudiando Modelos de Markov ya que sirven para predicción y se ajustaban a lo que necesitábamos.

### 5.10.3. Implementación del predictor

Lo primero que debemos comentar sobre la implementación es que después de buscar librerías que implementan este algoritmo observamos que no se adaptaban a lo que necesitábamos para resolver nuestro problema y debíamos realizar una modificación sustancial de las mismas para poder utilizarlas. Por lo que decidimos que tras estudiar el algoritmo era factible implemen-

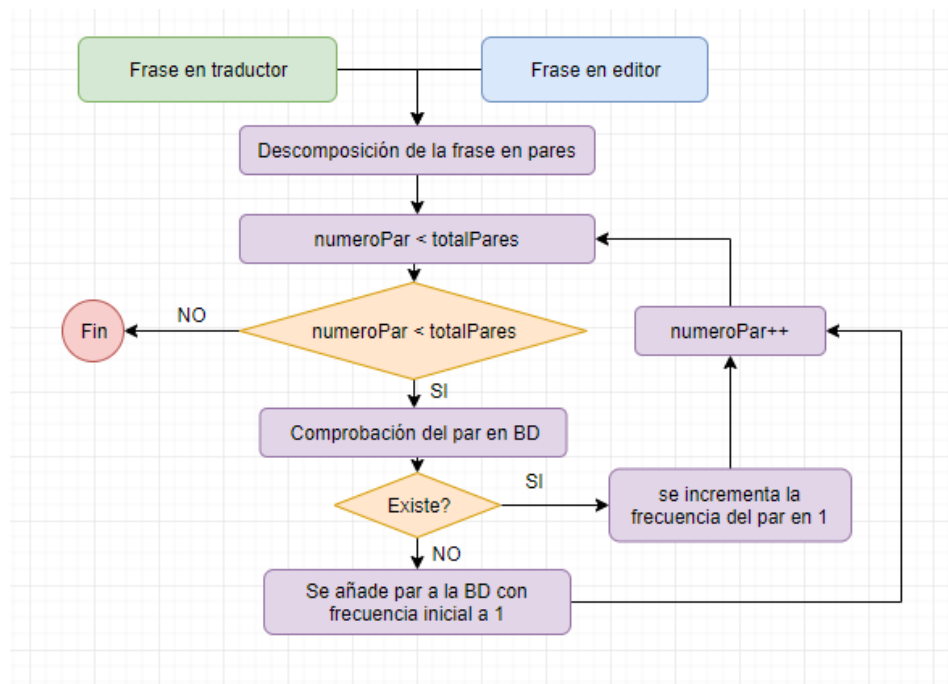


Figura 5.5: Diagrama representativo del predictor diseñado por Mónica

tarlo nosotros mismos eligiendo para resolver nuestro problema Cadenas de Markov, como ya hemos comentado anteriormente.

El lenguaje de programación elegido es java, ya que ambos miembros del equipo de proyecto lo conocen y tienen experiencia trabajando con el.

#### ■ Representación de los datos

Para definir el vector de probabilidad  $P$  para conocer la probabilidad de transición a cada estado en el paso siguiente se ha llevado a cabo mediante un array de  $n$  elementos en el que cada posición representa la probabilidad de transitar a un estado.

Para representar la matriz de transición la primera idea fue utilizar dos matrices de orden  $n$  en la que en una se almacenarán las probabilidades de transición entre cada par de estados y en la otra el número de transiciones entre cada par de estados. Esta idea fue rápidamente descartada al observar que la matriz que almacena las probabilidades no es necesaria y se puede sustituir por un vector que indica para cada estado las transiciones totales desde ese estado hacia cualquier otro y las probabilidades de transición se pueden calcular entre dos estados se dividiendo la posición de la matriz entre el elemento del vector que indica la fila de la matriz. Con este cambio se produce un ahorro de espacio en memoria y en tiempo al no ser necesario actualizar la matriz

entera constantemente.

El tamaño inicial de los vectores se ha decidido que sea 1000 ya que es un número suficientemente grande para evitar que se tenga que redimensionar el tamaño de los vectores con frecuencia al principio del uso del predictor que es cuando mas podría crecer al tener que incorporar nuevos estados. Posteriormente cuando ya se ha utilizado el predictor llega un punto que no se van a añadir prácticamente mas estados nuevos y por lo que si hay que ampliar el tamaño no será un problema como tener que redimensionar al principio de su uso que habría que hacerlo frecuentemente.

Para identificar los estados se ha creado un vector en el que se guarda el identificador de cada estado como un String.

#### ■ Principales funcionalidades implementadas

Las principales funcionalidades que se han implementado es que dado un estado te devuelva los estados mas probables a los que puede transitar en el paso siguiente. Para calcular las probabilidades de transición en el vector de probabilidad P, que indica la probabilidad de estar en cada estado se pone a 1 la probabilidad de estar en el estado dado y 0 en el resto, ya que conocemos el estado actual. Para realizar el producto del vector de probabilidad por la matriz de transición en este caso serviría con multiplicar el elemento del vector con probabilidad 1 por la fila de la matriz correspondiente, pero se ha implementado multiplicando todos los componentes del vector para que si se desea estimar las probabilidades de transición para dentro de mas de un paso se pueda realizar la operación sin realizar modificaciones, ya que de la forma descrita anteriormente no sería posible.

Hay que destacar el caso en el que se pida predicción para una palabra que se acaba de introducir en el sistema y no cuenta con ninguna transición. Para este caso concreto el algoritmo devuelve una probabilidad cero de transitar a cada uno de los estado, por lo que para que devuelva una predicción se ha asignado la misma probabilidad a cada posición del array de probabilidades. Esto se ha pensado para que el usuario tenga una predicción aunque desde el estado actual no se haya realizado una transición todavía.

Otra funcionalidad importante que se ha implementado es actualizar la matriz de transición después de cada cambio de estado. Esta funcionalidad sirve para que la matriz de transición se vaya modificando y la predicción vaya siendo mas precisa. Este método también sirve para entrenar al algoritmo realizando una llamada con el estado actual y el estado al que se va a transitar.

#### ■ Entrenamiento del predictor

Al resolver el problema de la predicción estuvimos estudiando si deberíamos entrenar o no el algoritmo, ya que si lo entrenamos podríamos dar desde un primer momento una predicción basándonos en composiciones que fueran mas comunes entre los usuarios a los que va dirigida la aplicación.

Por otra parte pensamos que cada usuario tiene sus costumbres y sus gustos y si entrenamos el algoritmo puede ser que no se adapte a él, además que si está entrenado tarda mas tiempo en adaptarse a los usuarios.

Por lo tanto hemos decidido que no vamos a entrenar el predictor, aunque como se ha indicado anteriormente hay un método que facilita esta funcionalidad. Esta decisión la hemos tomado porque creemos que es mas importante que el predictor se adapte mas rápidamente a los usuarios que dar una predicción generalizando a los usuarios y que no se adapte realmente a lo que necesitan además de que tarde mas en adaptarse al estar ya entrenado.

#### **5.10.4. Creación de servicios REST**

Para poder utilizar el predictor que hace uso de modelos de markov desde la aplicación web se ha utilizado la librería Jersey.

Como ya se ha comentado anteriormente la librería Jersey sirve para crear servicios REST desde clases Java. Nosotros hemos utilizado un tutorial <sup>3</sup> para comprobar como funciona esta librería y posteriormente lo hemos adaptado a nuestro predictor. Con esto hemos conseguido acceder a los métodos de nuestra clase Java que implementa el predictor como servicios web desde nuestra página implementada en PHP.

### **5.11. Conclusiones**

En este capítulo hemos hablado sobre la implementación de la página web, desgranando las principales áreas, las bases de datos de las que hemos expuesto las tablas que se han creado y las relaciones entre ellas y los algoritmos para la predicción indicando además como se han introducido en la página.

En el siguiente capítulo realizaremos las pruebas para comprobar el correcto funcionamiento de lo que hemos desarrollado, ver si se cumplen los objetivos de usabilidad que nos hemos propuesto además de encontrar errores para poder solventarlos.

---

<sup>3</sup><https://dzone.com/articles/rest-webservices-using-jersey-2x-and-maven-1>

# Capítulo 6

## Pruebas

*“No lo arruines. Los buenos planetas son difíciles de encontrar.”*

— Revista Time

En este capítulo vamos a realizar pruebas a nuestra aplicación para conocer si el trabajo desarrollado es realmente de utilidad para los usuarios finales, comprobar si se cumplen los requisitos de usabilidad y poder encontrar defectos y errores para corregirlos posteriormente.

### 6.1. Evaluación de la aplicación

El objetivo de esta fase es que los usuarios finales utilicen la aplicación para comprobar si se cumplen los objetivos de usabilidad que se pretenden conseguir, además de localizar errores para poder solventarlos y encontrar nuevas funcionalidades que se puedan añadir posteriormente.

Nuestra aplicación abarca diversas opciones y vamos a realizar una evaluación de cada una de ellas utilizando las 10 heurísticas de diseño de Nielsen (ver apartado 2.5.1.5). Para realizar esta evaluación se propondrá a los usuarios que realicen una serie de tareas que abarquen las funcionalidades de la aplicación, y que redacten en un formulario que vamos a crear, los problemas que han detectado para realizar cada tarea y nos cuenten sus impresiones con la aplicación. De esta forma podremos saber los inconvenientes que les puedan surgir a los usuarios al utilizar la aplicación para posteriormente eliminarlos.

Las tareas que se proponen a los usuarios son:

- Registrarse en la aplicación.
- Traducir una frase utilizando el traductor.
- Guardar la composición y consultarla en la pantalla **Mi perfil**.
- Volver a la pantalla principal y seleccionar un pictograma de una categoría.

- Buscar un pictograma con el buscador.
- Subir una imagen en la pantalla **Mi perfil**.

Los cuestionarios utilizados para realizar la evaluación, junto con las respuestas de los usuarios se encuentran disponibles en el repositorio de GitHub de este proyecto.

Después de probar la aplicación los usuarios nos han devuelto un documento indicando las ventajas y los problemas que habían detectado al realizar las tareas propuestas.

Cabe destacar que las personas encargadas de realizar las pruebas han sido seleccionadas en base a que cuentan con experiencia trabajando y tratando con personas con TEA o profesores que poseen conocimientos sobre aplicaciones enfocadas a niños.

### 6.1.1. Evaluación heurística

Tras obtener los resultados de los usuarios al realizar las pruebas de la aplicación vamos a proceder a analizarlos, exponiendo los problemas detectados e indicando las heurísticas que han sido violadas en cada caso. Posteriormente indicaremos cómo se podrían solucionar estos problemas.

- Nombre: Registrarse en la aplicación.
  - Problemas detectados:** Al tener que seleccionar la fecha en vez de poder insertarla de forma manual al usuario se le hace un poco pesado.
  - Heurística violada:** 3. Libertad y control por parte del usuario.
  - Solución:** Se podría implementar una funcionalidad que permita tanto seleccionar la fechas desde el menú desplegable como introducirla manualmente desde el teclado del dispositivo.
- Nombre: Traducir una frase utilizando el traductor.
  - Problemas detectados:** La traducción no presenta ningún problema. La única dificultad que se ha observado es que además del botón para traducir aparecen otros tres botones que generan confusión.
  - Heurística violada:** 5. Prevención de errores.
  - Solución:** Se podría dejar el botón de traducir solo en la parte izquierda y colocar el resto en la parte derecha o guardarlos dentro de un menú desplegable para evitar confusión al ofrecer demasiadas opciones.
- Nombre: Guardar la composición y consultarla en la pantalla **Mi perfil**.
  - Problemas detectados:** Al cambiar de pantalla no se observa a simple vista el apartado **Mis Frases** y hay que desplazarse por la página para verlo.
  - Heurística violada:** 6. Es mejor reconocer que recordar.



–**Solución:** se podría modificar la estructura de la pantalla **Mi Perfil** de modo que aparezcan todas las opciones o colocar el apartado **Mis Frases** ya que es probable que se utilice más que por ejemplo cambiar la contraseña o darse de baja.

- **Nombre:** Volver a la pantalla principal y seleccionar un pictograma de una categoría.

–**Problemas detectados:** No se ha detectado ningún problema al realizar la tarea pero se ha indicado que estaría bien que el usuario pudiera crear categorías propias. También debería aparecer el pictograma que define la categoría.

–**Heurística violada:** 3. Libertad y control por parte del usuario.

–**Solución:** Se podría implementar una funcionalidad que permita al usuario crear categorías propias e insertar en estas los pictogramas que desee, tanto propios como los proporcionados por el sistema. Además junto al nombre de la categoría podría aparecer el pictograma que la representa haciendo más visible para los usuarios el contenido de cada categoría.

- **Nombre:** Buscar un pictograma con el buscador.

–**Problemas detectados:** Aparecen imágenes que tienen poca relación con el concepto buscado. También ha aparecido la sugerencia de poder cambiar el lugar que ocupan los pictogramas dentro de la composición.

–**Heurísticas violadas:** 8. Diseño minimalista y 3. Libertad y control por parte del usuario.

–**Solución:** Se podría limitar el número de pictogramas que aparecen al realizar una búsqueda e implementar una funcionalidad que permita cambiar el orden de los pictogramas dentro de una composición.

- **Nombre:** Subir una imagen en la pantalla **Mi perfil**.

–**Problemas detectados:** Debería ofrecer la posibilidad de subir varios formatos de fotos, no solo en formato **png**. También se ha indicado que podría aclararse el tamaño máximo que puede ocupar la fotografía que se desea subir y ofrecer la posibilidad de aumentar el tamaño.

–**Heurísticas violadas:** 3. Libertad y control por parte del usuario, 9. Permitir al usuario solucionar el error.

–**Solución:** Se podría implementar la funcionalidad de ampliar las imágenes una vez que se han subido y la opción de subir más formatos de imágenes.

### 6.1.2. Aspectos positivos

Casi todos los usuarios que han realizado las pruebas a la aplicación y los que nos han hecho llegar sus opiniones a través del correo de PictoPad han destacado que es una herramienta muy útil a la hora de comunicarse con personas con TEA y otros problemas de comunicación.

Un aspecto destacado es que la traducción resulta muy útil para poder realizar una conversación teniendo apoyo visual.

También nos han comentado que además de ser de ayuda para niños con TEA también lo es para los padres de éstos permitiendo comunicarse de una forma mas sencilla con ellos.

Sobre la interfaz gráfica nos han comentado que es muy intuitiva y que facilita su uso por parte de los niños.

Muchos usuarios también nos han enviado opiniones positivas acerca de la predicción, comentando que en la mayoría de los casos es acertada.

Otra de las funcionalidades que ha sido destacada y que ha llamado la atención es la posibilidad de descargar las composiciones en PDF pudiendo recurrir a ellas cuando se necesiten.

## 6.2. Evaluación de los predictores

Teniendo en cuenta que nuestros usuarios son personas con TEA para realizar las pruebas de predicción hemos pensado que lo vamos a realizar con composiciones de pictogramas.

Para obtener las frases que hemos utilizado hemos recurrido a la web de ARASAAC ya que cuenta con un amplio catálogo de recursos basados en pictogramas de uso muy generalizado y pensados para las personas con TEA.

Para realizar las pruebas hemos seleccionado una colección de 26 frases obtenidas de los materiales ofrecidos por ARASAAC<sup>1</sup>, como ya hemos comentado antes para comprobar si la predicción se realiza correctamente.

### 6.2.1. Resultados

Después de llevar a cabo las primeras composiciones, se observa que al introducir una palabra que ya estaba en el predictor se devuelve la siguiente mas probable. Las primeras veces que se empieza a utilizar pueden variar las predicciones significativamente ya que se puede introducir una misma palabra seguida de varias diferentes en cada caso y al no haberse producido un número significativo de transiciones, al introducir una nueva, las probabilidades de la predicción pueden verse alteradas. Esto también sucede cuando hay palabras

---

<sup>1</sup>[http://www.arasaac.org/materiales.php?id\\_material=1307](http://www.arasaac.org/materiales.php?id_material=1307)

que se utilizan con poca frecuencia y al introducir una palabra nueva después de una que tiene poca frecuencia de uso se altera la predicción.

Según aumenta el uso del predictor se observa que la predicción se va ajustando con lo que se ha ido introduciendo anteriormente y no cambia si se usa una palabra nueva debido a que otra ya ha sido utilizada con mayor frecuencia, y al introducir la nueva no altera en gran medida la predicción hasta que esta sea usada mas repetidamente. Por ejemplo si después de **el** aparece repetidamente **conejo**, si posteriormente después de **el** aparece **perro**, el predictor seguirá dando mas probabilidad de aparecer a **conejo** que a **perro**.

Con el predictor de análisis de frecuencias al igual que con el predictor que utiliza Cadenas de Markov si después de una palabra dada hay 2 con la misma posibilidad de aparición muestra la última palabra que se ha añadido al predictor. Por ejemplo si utilizamos la frase **la sirena ve la mesa** y en el predictor no está **sirena** ni **mesa**, la siguiente vez que se introduzca **la** mostrará **mesa** como mas probable.

En el caso de que se utilice una palabra nueva después de una existente, si se introduce al principio de utilizar el predictor, puede aparecer mas fácilmente entre las palabras mas probables que si se introduce después de llevar mas tiempo utilizándolo. Esto puede suponer un ahorro de tiempo en el caso de que los usuarios suelen utilizar las mismas expresiones pero perjudicar cuando se utilicen expresiones diferentes ya que la predicción no será correcta hasta que haya transcurrido un mayor tiempo.

Cabe destacar que en el caso de intentar obtener una predicción para una palabra que se acaba de introducir en el sistema y no se ha producido todavía ninguna transición, el traductor basado en cadenas de Markov devuelve una predicción mientras que el predictor basado en análisis de frecuencias no devuelve predicción.

En nuestro caso conociendo las características de nuestros usuarios, que son personas con TEA y normalmente utilizan las mismas expresiones, es positivo que conozcan lo que va a suceder evitando los cambios bruscos. Por lo tanto creemos que es positivo para ellos que al introducir una palabra la predicción les ofrezca la posibilidad de utilizar a continuación otra que es la que ellos esperan.

## 6.3. Conclusiones

En este capítulo hemos realizado las pruebas de nuestra aplicación, tanto a la página web como a los predictores y a la usabilidad.

Tras la realización de las pruebas se ha comprobado que se cumplen los requisitos de usabilidad y se han obtenido resultados que identifican modificaciones que potencian la funcionalidad de la aplicación.

Estamos contentos de los resultados obtenidos tanto en el funcionamiento

de los predictores, ya que ambos funcionan correctamente en los casos en los que han sido probados, como de los resultados obtenidos de la interfaz, puesto que en este caso la impresión de los usuarios ha sido positiva y les ha parecido sencillo de usar y de una gran utilidad.

# Capítulo 7

## Trabajo individual

En este capítulo cada uno de los autores de este proyecto va a presentar el trabajo que ha realizado para que haya sido posible llevar a cabo este proyecto.

### 7.1. Trabajo realizado por Mónica

El arranque del proyecto comenzó por la memoria. Inicialmente nos dedicamos a buscar información sobre los usuarios para los que está orientada la aplicación, para así poder tener un mayor conocimiento de que necesidades íbamos a tener que satisfacer. Después, continuamos la investigación buscando aplicaciones que existiesen actualmente. De todas ellas cogimos ideas y pensamos en las carecias que tenían y cuales podrían ser aspectos a mejorar.

Una vez recopilamos todas las ideas, empezamos a pensar el diseño y las tecnologías que queríamos utilizar. Cada uno realizó un boceto de forma independiente, y después nos juntamos para compararlos y sacar la mejor versión posible. Junto al diseño inicial de la interfaz, creamos entre ambos integrantes un esquema de base de datos inicial. Con todo ello decidido dividimos en el trabajo, quedandose Iván con la parte del predictor y yo con el desarrollo de la web. A la vez que cada uno realizaba su parte, se iba completando la memoria.

Una vez dividido el trabajo yo comencé el diseño de la aplicación basándome en el boceto final. La implementación comenzó con el diseño de la página principal ya que es la que va a contener todas las funcionalidades. El diseño inicialmente lo desarrollé utilizando html, php y css, pero según se iba incrementando el número de componentes decidí aprender bootstrap y cambiar las hojas de estilos propias por las que ofrece bootstrap y así crear un estilo igualado en toda la página. Junto al cambio de estilo, también realicé una reorganización de la interfaz principal dividiendola en cuatro áreas bien

diferenciadas para que no quedase todo muy centrado y amontonado en un mismo sitio. Las dos principales áreas a realizar eran el traductor y el editor y por ello son las primeras que comencé a realizar. Durante su creación es cuando me dí cuenta que iba a tener que utilizar javascript, jquery y ajax para realizar algunas de las funciones como las de eliminar las imágenes, o hacer que no se recargue continuamente la página. A pesar de tener una idea base de estas tecnologías tuve que tomarme un tiempo para aprender a utilizarlas mejor para poder amoldarlas a las funcionalidades que buscaba implementar. Para la parte del traductor decidí utilizar el servicio de traducción de PICTAR que había desarrollado un antiguo alumno en el master.

Una vez conseguidas las funcionalidades prioritarias comencé a desarrollar la parte de las categorías. Para ello tuve que buscar en la página de ARASAAC los pictogramas, descargarlos y añadirlos a la base de datos. En este punto es donde llegó el primer cambio en la base de datos ya que cree una nueva tabla en la que guardar el nombre de los pictogramas junto a la categoría que creí que mejor definía la imagen. Finalmente hay un total de 15394 ya que solo he metido pictogramas en color. Con las categorías implementadas tomé la idea del buscador de PICTAR y también cree uno.

Para cumplir el objetivo de que la aplicación se pareciese a una agenda cree un registro e inicio de sesión para que así los usuarios tuviesen un perfil donde poder subir imágenes y así crear pictogramas personalizados. Con el perfil acabado, pensé en el futuro de la aplicación y en facilitar el trabajo a quien vaya a administrarla en el futuro ya seamos nosotros mismos u otra persona. Para ello he creado un panel de administración con el que poder realizar las tareas más necesarias para mantener la aplicación como es el añadir o eliminar usuarios o los pictogramas sin tener que acceder directamente al servidor ni a la base de datos.

Terminada toda la página y viendo que mi compañero tenía complicaciones para desarrollar el predictor decidí implementar uno en la propia aplicación basado en frecuencias. Aquí llegó otro de los cambios en base de datos, ya que añadí una nueva tabla que guardase los pares de palabras junto a la frecuencia con la que se utilizan juntas y que se vaya actualizando según los usuarios las utilicen en la página.

Cuando más o menos estaba todo nos juntamos y revisamos el trabajo junto al profesor y ahí salieron nuevas funcionalidades como la edición de frases ya utilizadas por los usuarios o la disminución de las opciones que ofrece el predictor, entre otras mejoras.

Con una versión estable de la aplicación intenté hacer funcionar el predictor de mi compañero en el servidor, para poder añadirlo a la aplicación pero por mucho que lo intentamos ambos, no conseguimos hacerlo correr en el servidor, y por ello se ha mantenido el basado en frecuencias.

Para recibir feedback sobre la aplicación envié, al igual que mi compañero, el enlace para que lo probasen diversos usuarios, algunos de ellos con

parientes con trastorno del espectro autista. También ha sido enviada a algunos profesores que se relacionan con niños con TEA. Para recopilar las respuestas cree un correo electrónico, que aparece en el pie de página de la aplicación y una encuesta con las tareas para la evaluación que creo mi compañero Iván.

Con todo ello acabado nos hemos dedicado a repasar continuamente la memoria y a corregir los errores que íbamos revisando.

## 7.2. Trabajo realizado por Iván

Al comienzo del proyecto la primera tarea que realizamos fue investigar el Estado de la Cuestión, comenzando a redactar la memoria. En este capítulo ambos realizamos la parte relacionada con el Trastorno del Espectro Autista, buscando información acerca de los diferentes trastornos existentes. Posteriormente investigamos los beneficios de los SAACs en las personas con TEA, en particular los pictogramas que eran los que nos interesaban para el desarrollo de nuestro proyecto. Posteriormente investigamos las aplicaciones basadas en pictogramas para conocer las funcionalidades que poseen.

Como se había decidido implementar un predictor estuve investigando los mecanismos que se utilizan actualmente para realizar predicciones y cuáles se podrían adaptar mejor a nuestro problema. Una vez que se decidió utilizar modelos de Markov, estudié las diferentes implementaciones del algoritmo llevando a cabo la elección de la cadena de Markov ya que era el que mejor resolvía nuestro problema.

Posteriormente estuvimos investigando la metodología del Diseño Centrado en el Usuario porque era la que nos parecía mas beneficiosa para nuestro proyecto ya que se centra en las personas que van a hacer uso de la aplicación.

En el diseño de la interfaz gráfica realizamos primero un diseño competitivo sin enseñar el diseño al otro componente del grupo. Cada uno de nosotros realizamos un diseño individualmente y posteriormente lo pusimos en común extrayendo las ideas mas interesantes de ambos diseños para crear la interfaz en común.

Llegados a este punto diseñamos un primer modelo para la base de datos, que posteriormente sería modificado por Mónica para adaptarlo a las necesidades de la página web.

En este punto ya había decidido que algoritmo iba a utilizar y comencé a buscar bibliotecas que lo implementaran para poder utilizarlas y poder dedicar tiempo a otras tareas. Como las bibliotecas que encontré eran en su mayoría implementaciones destinadas a generar texto automáticamente o requerían una modificación sustancial para poder utilizarlas, decidí implementar el algoritmo y elegir la representación de los datos que me parecía mas apropiada.

Tras esto comencé a investigar bibliotecas que permitieran crear servicios

REST para poder hacer llamadas al algoritmo desde la página web. Con la biblioteca **Jersey** cree los servicios que permitían obtener la predicción para una palabra y actualizar el predictor cuando se ha producido una transición.

Tras tener el predictor funcionando y Mónica haber desarrollado otro basado en análisis de frecuencia estuve realizando pruebas de predicción para comprobar cómo se comportan ambos predictores. Para probarlos seleccioné un libro de frases de la web de ARASAAC ya que quería probarlos con datos que sean lo mas parecidos posibles a los que utilizan los usuarios. Una vez que realicé las pruebas documenté en la memoria las conclusiones obtenidas.

Una vez que la página estaba terminada diseñé la evaluación heurística, eligiendo las tareas que se iban a proponer que realizaran los evaluadores y creando un documento para que fuera mas sencillo realizar la evaluación. Cabe destacar que Mónica realizó un formulario web para facilitar la tarea a los evaluadores si no utilizaban un ordenador.

Después de realizar la evaluación recogí los datos e interpreté los resultados indicando los problemas detectados, identificando las heurísticas violadas con cada problema y una propuesta de solución para los errores indicados. Posteriormente añadí a la memoria los resultados obtenidos.

Finalmente tras añadir los resultados de las evaluaciones procedía a realizar las conclusiones de la memoria haciendo balance de los objetivos, indicando de si se habían cumplido y como. Tras esto tanto Mónica como yo indicamos cuáles podrían ser las líneas de investigación dejándolas redactadas en la memoria como trabajo futuro.



## Capítulo 8

# Conclusiones y Trabajo Futuro

En este capítulo vamos a exponer las conclusiones sobre este trabajo, analizando si se han cumplido los objetivos expuestos para este proyecto anteriormente. También se indicaran que líneas de trabajo se podrían seguir en un futuro para seguir con el desarrollo de este proyecto e ir mejorándolo paulatinamente.

### 8.1. Conclusiones

El principal objetivo de este trabajo era crear una aplicación web que resultase accesible y facilitase la comunicación a personas con TEA. Para alcanzarlo, por un lado era necesario proporcionar un traductor de texto a pictogramas que ayudase a la construcción y comprensión de oraciones por parte de los individuos con TEA. Por otro lado, se buscaba crear un traductor que realizase la función contraria. Con ello se pretendía facilitar la comunicación entre personas autista con su entorno social. Ciertamente es que ya existen aplicaciones con estas funciones, por ello se trataba de mejorar ambos traductores y para ello se propuso realizar un predictor basado en modelos de Markov, que fuese utilizable por ambas opciones de traducción.

Para alcanzar esta meta se ha desarrollado una aplicación web que sigue las pautas en las que se basa el diseño centrado en el usuario además de tener en cuenta las necesidades que presentan las personas con TEA. Para comprobar si el diseño cumple las expectativas con las que ha sido creado, ha sido evaluado por diversos usuarios. Ya que las evaluaciones no las hemos podido realizar con usuarios reales, hemos buscado personas que o bien trabajen con personas con TEA o tengan a alguien en su entorno más cercano como son profesores o familiares.

Respecto a la creación del predictor basado en modelos de Markov, se ha desarrollado y creado un servicio rest para acceder a él, el cual solo hemos

podido probar de forma local. Se ha comprobado que funciona correctamente pero no se ha conseguido ejecutar en el servidor. Para realizar las predicciones hemos utilizado el otro predictor basado en análisis de frecuencias tras comprobar que también proporciona una predicción correcta y bastante similar al anterior. A pesar de que no se ha podido completar el objetivo del predictor, se ha optado por buscar una alternativa que realice la misma función.

Otro de los objetivos de este trabajo era demostrar y aplicar los conocimientos adquiridos en las asignaturas cursadas durante el grado de ingeniería informática. Entre todas las asignaturas destacamos las siguientes, puesto que son las que mayor ayuda nos han aportado:

- Bases de datos : Que nos ha ayudado a crear la base de datos MySQL
- Tecnología de la Programación : Nos ha ayudado a descomponer los problemas de programación y diseñar soluciones para ellos.
- Estructura de Datos y Algoritmos: usada para pensar de forma eficiente tanto en la elección como en la representación de los datos cuando han sido diseñados.
- Ética, Legislación y Profesión : nos ha permitido conocer las licencias utilizadas actualmente, la normativa referente al tratamiento de datos y cómo usar el software libre.
- Aplicaciones Web: nos ha servido para el desarrollo de toda la aplicación ya que en ella se aprende la base de los lenguajes web más utilizados los cuales se utilizan para todo el desarrollo de la aplicación.
- Ampliación de Bases de Datos: en esta asignatura se enseñan bases de datos no relacionales. A pesar de que no se han utilizado, nos ha ayudado a pensar que tipo de base de datos sería más apropiada para este proyecto.

Gracias a este proyecto hemos aprendido a gestionar el tiempo de las diferentes tareas a realizar. También nos ha hecho ser más autodidactas e independientes a la hora de hacer elecciones respecto a los diseños. Nos ha enseñado también como es el trabajo en equipo en el que no hay un supervisor al mando.

En conclusión, este proyecto ha hecho que desarrollemos facetas importantes que en el futuro nos pueden aportar cualidades necesarias para nuestra incorporación en el mundo laboral.

## 8.2. Trabajo Futuro

Una vez que terminado el proyecto, hemos hecho balance de los objetivos propuestos inicialmente y hemos observado que los hemos cumplido en su gran mayoría pero no en su totalidad. Por otra parte, tras realizar las distintas evaluaciones, hemos obtenido nuevos objetivos que podrían llevarse acabo en el futuro.

A continuación, enumeraremos los objetivos que creemos que nos han quedado pendiente, además de todo el trabajo futuro posible:

- Se podría tratar de simplificar la zona del traductor, cambiando la distribución de los botones de **Guardar**, **Descargar** y **Limpiar** u ocultándolos en un menú desplegable. Otra opción podría ser identificar cada botón con un pictograma además de con el texto escrito.
- Permitir elegir las fechas en las que se quieren guardar las traducciones, para que actue como una agenda planificadora
- Abrir la categoría que tienen los usuarios *Mis pictogramas* a crear sus propias categorías asociadas al usuario permitiendo dar una mayor personalización a la búsqueda. También se podría representar cada categoría con una pictograma además de con el texto.
- Permitir al usuario jugar con el orden de os pictogramas cuando está realizando una edición.
- Permitir que el usuario pueda variar el tamaño de las imágenes tanto de las que sube como de los pictogramas de la aplicación.
- Actualmente se permiten subir imágenes tan solo en formatos **png** y **jpg**, por lo que se ha pensado que se podría dar la posibilidad de subir imágenes en otros formatos.
- Como hemos comentado en la sección anterior no hemos podido utilizar el predictor basado en modelos de Markov. Éste se encuentra ya desarrollado y probado al igual que se han creado los servicios REST necesarios para su uso, por lo tanto sería interesante poder subir este predictor al servidor y ponerlo en la aplicación.
- Dado que hemos estado trabajando con SAACs utilizando los pictogramas, creemos que sería interesante poder aumentar la funcionalidad de esta aplicación para que pueda ser utilizada por personas que tienen discapacidad visual. Para ello hemos pensado que sería beneficioso poder añadir reconocimiento por voz a la aplicación. Actualmente existen bibliotecas que implementan esta funcionalidad como por ejemplo API Cloud Speech<sup>1</sup> desarrollada por Google.

---

<sup>1</sup><https://cloud.google.com/speech/?hl=es>

- Actualmente en la base de datos solo se encuentran pictogramas a color, por lo que sería útil añadir pictogramas en blanco y negro y esquemáticos para las personas que hacen uso de este tipo de pictogramas. También se podría implementar una opción que permita que los pictogramas que aparezcan por defecto sean de un determinado tipo para evitar que el usuario tenga que estar seleccionando el tipo de pictograma cada vez que realiza una composición.
- Ya que en ARASAAC, que es la página de donde se han sacado los pictogramas, aparecen pictogramas en diferentes lenguajes, sería una buena idea poder internacionalizar la aplicación para llegar a mayor número de usuarios.

Dado el carácter social que tiene esta aplicación, nos gustaría continuar trabajando en ella pese a que se haya cerrado una fase con la presentación del proyecto. Con ello pretendemos evolucionarla y proporcionar un mayor valor y ajuste a las necesidades de los usuarios finales, de forma que continúen haciendo uso de ella y así nosotros consigamos el verdadero y principal objetivo de todo este trabajo, que es el de poder ayudar a la integración de las personas con TEA.

## Conclusions and Future Work

In this chapter we will present the conclusions of this work, analysing whether the objectives set out for this project above have been fulfilled. We also indicate which lines of work could be followed in the future to continue with the development of this project and gradually improve it.

### 8.1. Concluding Remarks

The main objective of this work was to create a web application that would be accessible and facilitate communication to people with ASD. To achieve this goal, on the one hand it was necessary to provide a text to pictogram translator that would help the construction and comprehension of sentences by individuals with ASD. On the other hand, the aim was to create a translator to perform the opposite function. The aim was to facilitate communication between autistic people and their social environment. It is true that there are already applications with these features, so it was a question of improving both translators and for this purpose it was proposed to make a predictor based on Markov models, which could be used by both translation options.

To achieve this goal, a web application has been developed that follows the guidelines on which user-centered design is based as well as taking into account the needs of people with ASD. In order to check whether the design fulfils the expectations with which it has been created, it has been evaluated by several users. Since we have not been able to carry out the evaluations with real users, we have looked for people who either work with people with ASD or have someone in their closest environment such as teachers or relatives.

Regarding the creation of the predictor based on Markov models, a rest service has been developed and created to access it, which we have only

been able to test locally. It has been verified that it works correctly but has not been able to run on the server. To make the predictions we have used the other predictor based on frequency analysis after verifying that it also provides a correct prediction and quite similar to the previous one. Although it has not been possible to complete the goal of the predictor, it has been opted to look for an alternative that performs the same function.

Another of the objectives of this work was to demonstrate and apply the knowledge acquired in the subjects studied during the degree in computer engineering. Among all the subjects we highlight the following, since they are the ones that have given us the most help:

- Databases : That helped us to create the MySQL database
- Programming Technology : This has allowed us to break down programming problems and design solutions for them.
- Data Structure and Algorithms: used to think efficiently about both the choice and representation of data when it has been designed.
- Ethics, Legislation and Profession : has allowed us to know the licenses currently used, the regulations concerning data processing and how to use free software.
- Web Applications: has served us for the development of the whole application as it learns the basis of the most used web languages which are used for the entire development of the application.
- Database Expansion: in this subject non-relational databases are taught. Although they have not been used, it has helped us to think what type of database would be more appropriate for this project.

Thanks to this project we have learned how to manage the time of the different tasks to be performed. It has also made us more self-taught and self-reliant when making design choices. It has also taught us what teamwork is like in which there is no supervisor in charge.

In conclusion, this project has led us to develop important facets that in the future may provide us with the qualities necessary for our incorporation into the world of work.

## 8.2. Future Work

Once the project was completed, we took stock of the objectives initially proposed and observed that we had achieved the vast majority but not all of them. On the other hand, after carrying out several evaluations, we have reached new objectives that could be implemented in the future.

We will now list the goals that we think we have yet to achieve, as well as all possible future work:

- The translator area could be simplified by changing the layout of the buttons for **Save**, **Download** and **Clean** or hiding them in a drop down menu. Another option could be to identify each button with a pictogram in addition to the typed text.
- Allow users to choose the dates on which you want to save the translations, so that it acts as a planning agenda.
- Open the category that users have *My pictograms* to create their own categories associated with the user allowing greater customization of the search. Each category could also be represented with a pictogram in addition to the text.
- Allow the user to play with the order of the pictograms when making an edition.
- Enable users to vary the size of both uploaded images and application pictograms.
- Images are currently allowed to be uploaded only in `textttpng` and `textttjpg` formats, so it has been thought that it might be possible to upload images in other formats.
- As we commented in the previous section we could not use the predictor based on Markov models. This is already developed and tested as well as REST services have been created necessary for its use, so it would be interesting to be able to upload this predictor to the server and put it in the application.
- Since we have been working with SAACs using pictograms, we think it would be interesting to be able to increase the functionality of this application so that it can be used by visually impaired people. We thought it would be beneficial to be able to add voice recognition to the application. Currently there are libraries that implement this functionality such as API Cloud Speech<sup>1</sup> developed by Google.
- Currently in the database are only color pictograms, so it would be useful to add black and white pictograms and schematics for people who make use of this type of pictograms. An option could also be implemented that allows the pictograms that appear by default to be of a certain type to avoid the user having to be selecting the type of pictogram each time he makes a composition.

---

<sup>1</sup><https://cloud.google.com/speech/?hl=es>

- As in ARASAAC, which is the page from which the pictograms have been taken, pictograms appear in different languages, it would be a good idea to internationalize the application to reach a greater number of users.

Given the social nature of this application, we would like to continue working on it despite the fact that a phase has been closed with the presentation of the project. With this we intend to evolve it and provide greater value and adjustment to the needs of end users, so that they continue to make use of it and so we get the real and main objective of all this work, which is to be able to help the integration of people with ASD.



# Bibliografía

*Y así, del mucho leer y del poco dormir,  
se le secó el cerebro de manera que vino  
a perder el juicio.*

Miguel de Cervantes Saavedra

¿que son los web services y quetecnologia usar en su desarrollo? <https://www.arsys.es/blog/programacion/disenio-web/web-services-desarrollo/>, 2015. [Internet, descargado 3-septiembre-2019].

¿que son los web services y quetecnologia usar en su desarrollo? <https://www.arsys.es/blog/programacion/disenio-web/web-services-desarrollo/>, 2015. [Internet, descargado 3-septiembre-2019].

Pictoagenda. <https://www.pictoagenda.com/>, 2018a. [Online; accessed 29-November-2018].

Pictotraductor. <https://www.pictotraductor.com/>, 2018b. [Online; accessed 29-November-2018].

¿qué es gnu/linux? <https://www.debian.org/releases/stretch/armel/ch01s02.html.es>, 2019. [Internet, descargado 3-septiembre-2019].

Usabilidad. <https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>, 2019. [Online; accessed 15-April-2019].

ANTONIO SÁNCHEZ RUIZ-GRANADOS, B. F. M., RAQUEL HERVÁS BALLESTEROS. Diseño de sistemas interactivos. tema 0. 2017.

APACHE. Apache. <https://httpd.apache.org/>, 2019. [Internet; descargado 19-marzo-2019].

B., G. Qué es jquery. <https://www.hostinger.es/tutoriales/que-es-jquery/>, 2019. [Internet, descargado 3-septiembre-2019].

- BASIL, C. ¿qué son los sistemas aumentativos y alternativos de comunicación (saac)? <http://www.arasaac.org/aac.php>, 2018. [Online; accessed 29-November-2018].
- BENITES, A. G. ¿qué es bootstrap? <https://devcode.la/blog/que-es-bootstrap/>, 2018. [Internet; descargado 3-septiembre-2019].
- CAULES, C. A. Introducción a servicios rest. <https://www.arquitecturajava.com/servicios-rest/>, 2013. [Online; accessed 19-March-2019].
- CONFEDERACIÓN AUTISMO ESPAÑA. Diseño para todas las personas de las tic. <http://www.autismo.org.es/sobre-los-TEA/accesibilidad>, 2018. [Online; accessed 29-November-2018].
- DELGADO, C. I. Mi comunicador de pictogramas. 2012.
- DOMÍNGUEZ, P. ¿qué es jquery? <https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery/4309498-que-es-jquery>, 2018. [Internet, descargado 3-septiembre-2019].
- EGUÍLUZ PÉREZ, J. Introducción a javascript. 2012.
- GRUPO DE TRABAJO DE LA GUÍA DE PRÁCTICA CLÍNICA PARA EL MANEJO DE PACIENTES CON TRASTORNOS DEL ESPECTRO AUTISTA EN ATENCIÓN PRIMARIA. Guía de práctica clínica para el manejo de pacientes con trastornos del espectro autista en atención primaria. [http://www.guiasalud.es/GPC/GPC\\_462\\_Autismo\\_Lain\\_Entr\\_resum.pdf](http://www.guiasalud.es/GPC/GPC_462_Autismo_Lain_Entr_resum.pdf), 2009. [Online; accessed 29-November-2018].
- GUTIÉRREZ, E. G. ¿quÉ es y para quÉ sirve html? 2019. [Internet; descargado 17-abril-2019].
- HASSAN-MONTERO, Y. y ORTEGA-SANTAMARÍA, S. *Informe APEI sobre usabilidad*, vol. 3. APEI, Asociación Profesional de Especialistas en Información, 2009.
- HERNANDEZ, U. Mvc (model, view, controller) explicado. <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>, 2019. [Internet; descargado 15-julio-2019].
- JAUME GUILERA. Síndrome de Asperger : ¿Qué es la Tríada de Wing? 2016.
- JURAFSKY, D. y MARTIN, J. H. Language modeling with ngrams. *Speech and Language Processing, August*, vol. 7, 2017.
- LUJÁN-MORA, S. *Programación en Internet: clientes web*. Editorial Club Universitario, 2001.

- MARÍN DIAZARAQUE, J. Cadenas de markov. *Departamento de Estadística, Universidad Carlos III de Madrid. Sitio web: <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/PEst/tema4pe.pdf>*, 2012.
- MARTÍN GUERRERO, A. *PICTAR: una herramienta de elaboración de contenido para personas con TEA basada en la traducción de texto a pictogramas*. Proyecto Fin de Master, Facultad de Informática - UCM, 2018.
- MEHDI ACHOUR, A. D., FRIEDHELM BETZ. Php. <https://www.php.net/>, 2019. [Internet; descargado 19-marzo-2019].
- MÉNDEZ, G., HERVÁS, R., BAUTISTA, S. y GALVÁN, P. Pictoeditor: un editor predictivo basado en pictogramas para facilitar la interacción con personas con discapacidad. En *XIX International Conference on Human-Computer Interaction (Interacción 2018)*. 2018.
- MIRÓ, A. ¿qué es y para que sirve github? <https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-github>, 2017. [Internet; descargado 16-julio-2019].
- NIELSEN, J. Ten usability heuristics. 2005.
- ORACLE. Mysql. <http://www.mysql.com/>, 2019. [Internet; descargado 19-marzo-2019].
- PAZ, L. y SZYSZLICAN, M. El diseño centrado en el usuario y su impacto en la identidad de los proyectos. *No Solo Usabilidad*, (13), 2014.
- SCHULZ, R. G. *Diseño web con CSS*. Marcombo, 2008.
- UNIVERSIDAD INTERNACIONAL DE VALENCIA. El uso de tics en escolares diagnosticados de autismo. <https://www.universidadviu.es/el-uso-de-tics-en-escolares-diagnosticados-de-autismo/>, 2014. [Online; accessed 29-November-2018].
- WIKIPEDIA. N-grama — wikipedia, la enciclopedia libre. 2018a. [Internet; descargado 29-noviembre-2018].
- WIKIPEDIA. Pictograma — wikipedia, la enciclopedia libre. 2018b. [Internet; descargado 29-noviembre-2018].
- WIKIPEDIA. Xampp — wikipedia, la enciclopedia libre. 2018c. [Internet; descargado 18-marzo-2019].
- WIKIPEDIA. Sql — wikipedia, la enciclopedia libre. 2019. [Internet; descargado 19-marzo-2019].

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –  
Bien podrán los encantadores quitarme la ventura,  
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero  
Don Quijote de la Mancha  
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.  
–No es menester firmarla – dijo Don Quijote–,  
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero  
Don Quijote de la Mancha  
Miguel de Cervantes*

